Classification Building Block: Maxent/Logistic Regression/Log-linear

CMSC 473/673

Frank Ferraro

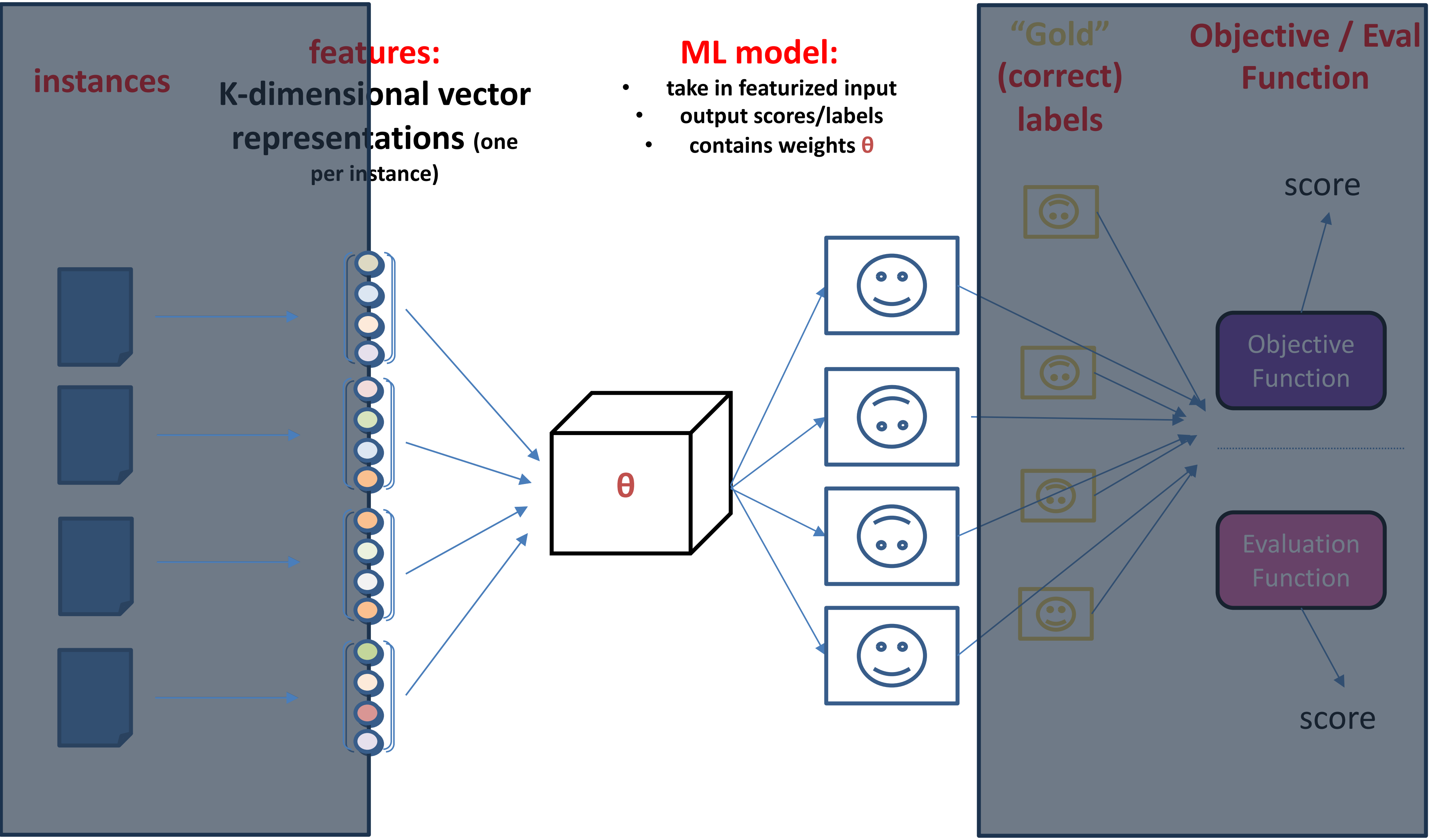
Some slides adapted from 3SLP

Outline

Maxent/Logistic Regression/Log-linear

- Defining the model
- Defining the objective
- Learning: Optimizing the objective
- Math: gradient derivation (advanced)

Defining the Model



instances

features: K-dimensional vector representations (one per instance)

ML model:

- take in featurized input
- output scores/labels
- contains weights θ

“Gold” (correct) labels

Objective / Eval Function

Terminology

common NLP term	Log-Linear Models
as statistical regression	(Multinomial) logistic regression
	Softmax regression
based in information theory	Maximum Entropy models (MaxEnt)
a form of	Generalized Linear Models
viewed as	Discriminative Naïve Bayes
to be cool today :)	Very shallow (sigmoidal) neural nets

Maxent Models are Flexible

Maxent models can be used:

- to design discriminatively trained classifiers, or
- to create featureful language models

- (among other approaches in NLP and ML more broadly)

Reminder!

Examining Assumption 3 Made for Classification Evaluation

- Given X, our classifier produces a score for each possible label

p(●|X) vs. p(○|X)

- Normally (*but this can be adjusted!)

$$\text{best label} = \arg\max_{\text{label}} P(\text{label}|\text{example})$$

Terminology: Posterior Probability

- Posterior probability:

p(●|X) vs. p(○|X)

- These *are* conditional probabilities
  - If ● and ○ are the only two options:

p(●|X) + p(○|X) = 1

  - and

p(●|X) ≥ 0, p(○|X) ≥ 0

Terminology (with variables)

- Posterior probability:

$$p(Y = \text{label}_1 \,|X) \text{ vs. } p(Y = \text{label}_0 \,|X)$$

- These *are* conditional probabilities

$$p(Y = \text{label}_1 \mid X) + p(Y = \text{label}_0 \mid X) = 1$$

$$p(Y = \text{label}_1 \mid X) \geq 0,$$
$$p(Y = \text{label}_0 \mid X) \geq 0$$

💡 Key Take-away 💡

We will ***learn*** this

$$\boldsymbol{p}(\boldsymbol{Y} \mid \boldsymbol{X})$$

Maxent Models for Classification: Discriminatively …

Directly model the posterior

$$p(Y \mid X) = \mathbf{maxent}(X; Y)$$

Discriminatively trained classifier

Maxent Models for Classification: Discriminatively *or* Generatively Trained

Directly model the posterior

$$p(Y \mid X) = \mathbf{maxent}(X; Y)$$

Discriminatively trained classifier

Model the posterior with Bayes rule

$$p(Y \mid X) \propto \mathbf{maxent}(X \mid Y) p(Y)$$

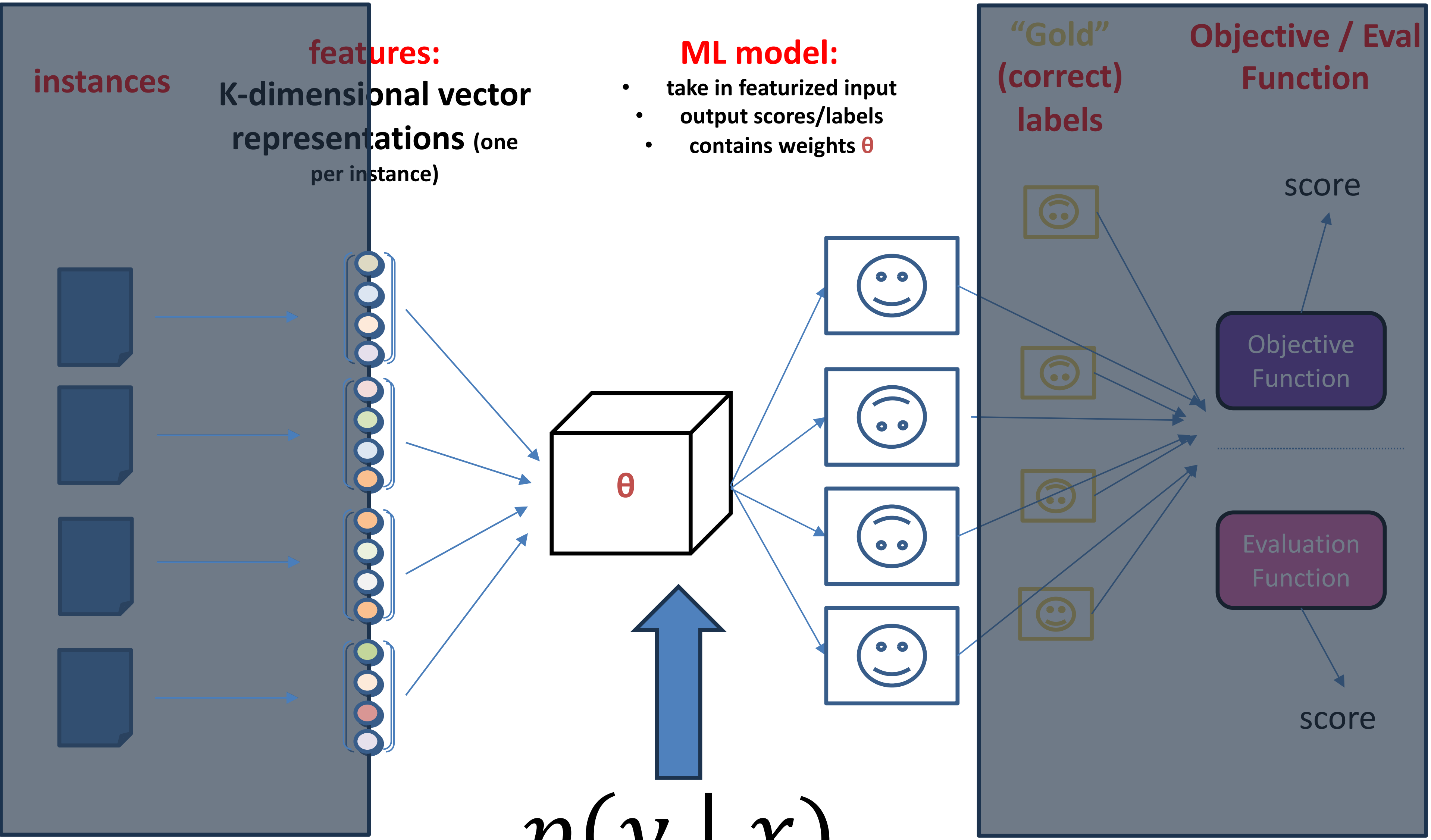
Generatively trained classifier with maxent-based language model

Maximum Entropy (Log-linear) Models For Discriminatively Trained Classifiers

(we'll start with this one)

$$p(y \mid x) = \mathrm{maxent}(x, y)$$

discriminatively trained:
classify in one go



instances

features:

K-dimensional vector representations (one per instance)

ML model:

- take in featurized input
- output scores/labels
- contains weights θ

"Gold" (correct) labels

Objective / Eval Function

$$p(y \mid x) = \mathrm{maxent}(x, y)$$

Core Aspects to Maxent Classifier p(y|x)

We need to define

- **features** $f(x)$ from x that are meaningful;
- **weights** θ (at least one per feature, often one per feature/label combination) to say how important each feature is; and
- a way to **form probabilities** from f and θ

Reminder!

Discriminative ML Classification in 30 Seconds

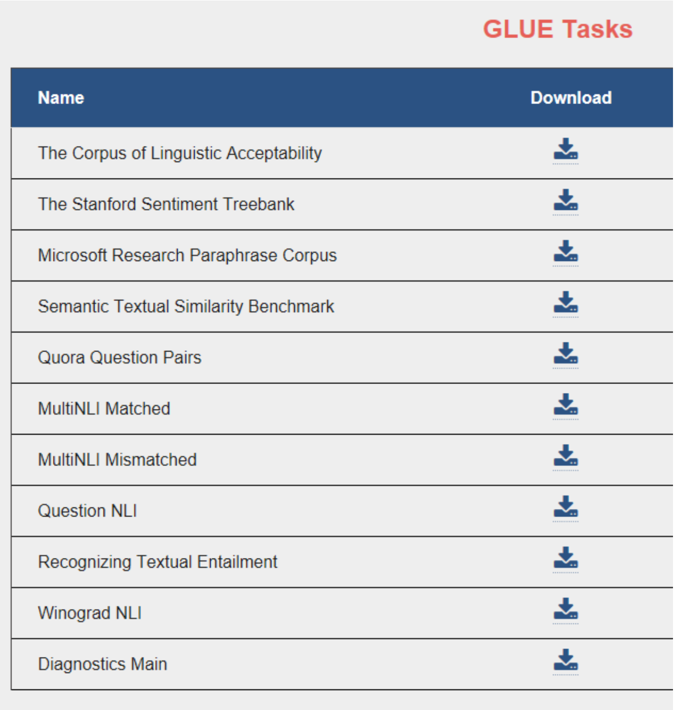
- Common goal: probabilistic classifier p(y | x)
- Often done by defining **features** between x and y that are meaningful
  - Denoted by a general vector of K features

$$f(x) = (f_1(x), \ldots, f_K(x))$$

- **Features can be thought of as "soft" rules**
  - **E.g., POSITIVE sentiments tweets *may* be more likely to have the word "happy"**

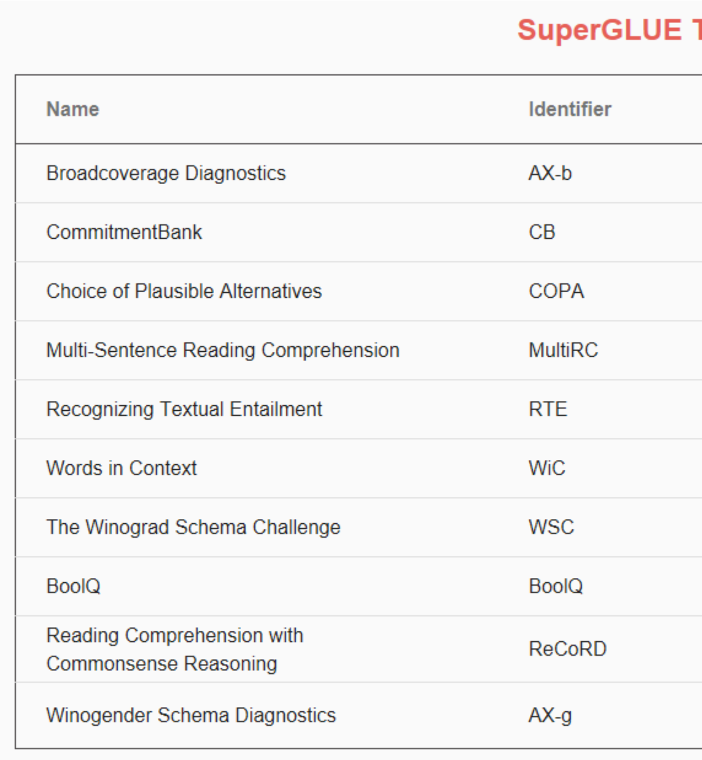
Example Classification Tasks

GLUE
https://gluebenchmark.com/
🤗datasets: glue



GLUE Tasks

Name	Download
The Corpus of Linguistic Acceptability	⬇
The Stanford Sentiment Treebank	⬇
Microsoft Research Paraphrase Corpus	⬇
Semantic Textual Similarity Benchmark	⬇
Quora Question Pairs	⬇
MultiNLI Matched	⬇
MultiNLI Mismatched	⬇
Question NLI	⬇
Recognizing Textual Entailment	⬇
Winograd NLI	⬇
Diagnostics Main	⬇



SuperGLUE T

Name	Identifier
Broadcoverage Diagnostics	AX-b
CommitmentBank	CB
Choice of Plausible Alternatives	COPA
Multi-Sentence Reading Comprehension	MultiRC
Recognizing Textual Entailment	RTE
Words in Context	WiC
The Winograd Schema Challenge	WSC
BoolQ	BoolQ
Reading Comprehension with Commonsense Reasoning	ReCoRD
Winogender Schema Diagnostics	AX-g

SuperGLUE

https://super.gluebenchmark.com/
🤗datasets: super_glue

Recognizing Textual Entailment (RTE)

Given a premise sentence s and hypothesis sentence h, determine if h “follows from” s

ENTAILMENT (yes):

NOT ENTAILED (no):

Recognizing Textual Entailment (RTE)

Given a premise sentence s and hypothesis sentence h, determine if h “follows from” s

ENTAILMENT (yes):

s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.

h: The Bulls basketball team is based in Chicago.

NOT ENTAILED (no):

Recognizing Textual Entailment (RTE)

Given a premise sentence s and hypothesis sentence h, determine if h "follows from" s

ENTAILMENT (yes):

s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.

h: The Bulls basketball team is based in Chicago.

NOT ENTAILED (no):

s: Based on a worldwide study of smoking-related fire and disaster data, UC Davis epidemiologists show smoking is a leading cause of fires and death from fires globally.

h: Domestic fires are the major cause of fire death.

RTE

s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.
h: The Bulls basketball team is based in Chicago.

ENTAILED

p(ENTAILED | s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.
h: The Bulls basketball team is based in Chicago.)

Discriminative Document Classification

s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.

h: The Bulls basketball team is based in Chicago.

ENTAILED

Discriminative Document Classification

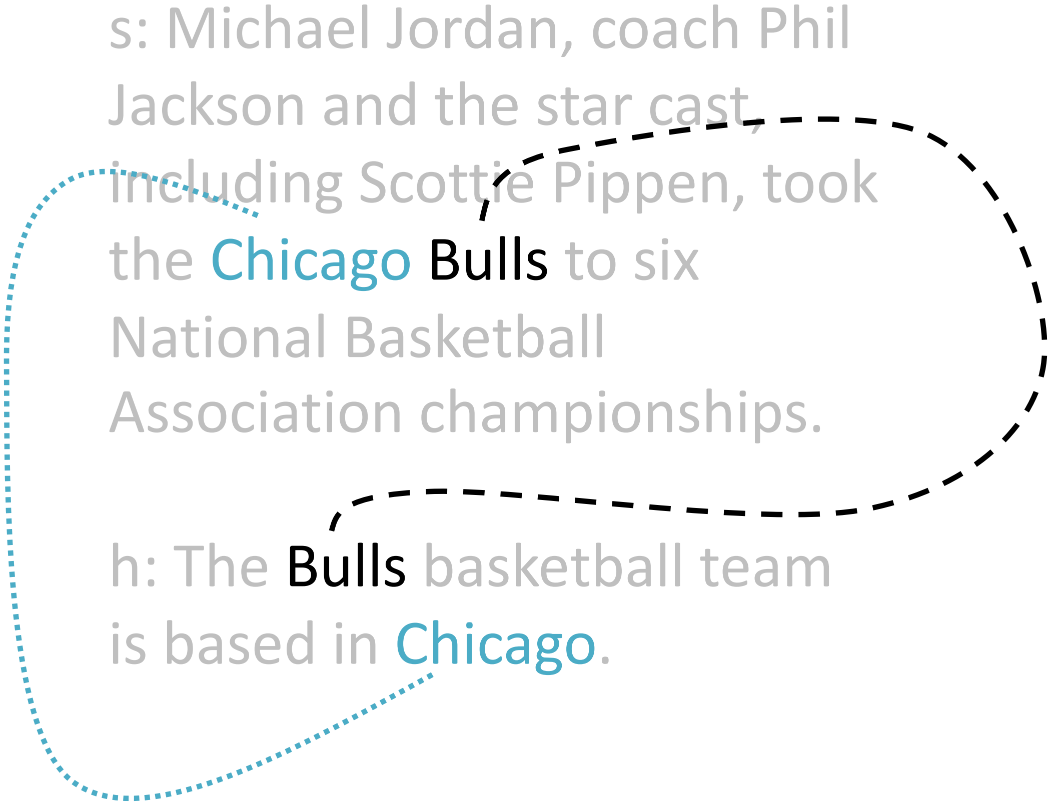
s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.

h: The Bulls basketball team is based in Chicago.

ENTAILED

These extractions are all **features** that have **fired** (likely have some significance)

Discriminative Document Classification



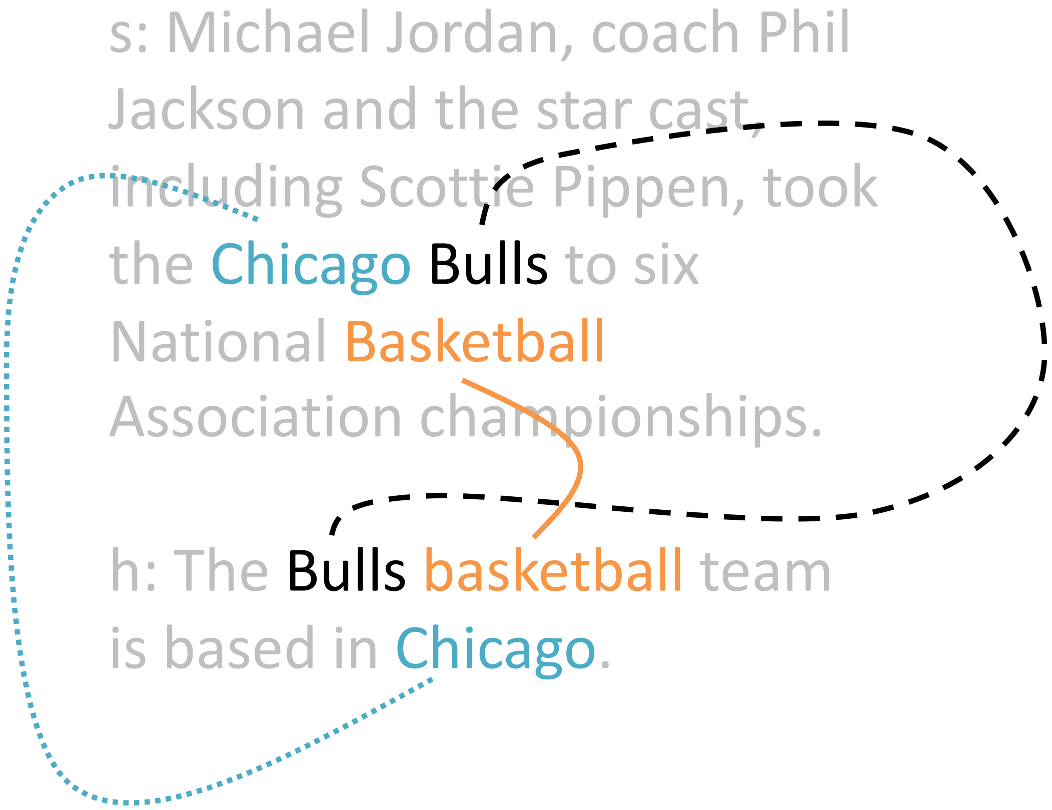
s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.

h: The Bulls basketball team is based in Chicago.

ENTAILED

These extractions are all **features** that have **fired** (likely have some significance)

Discriminative Document Classification



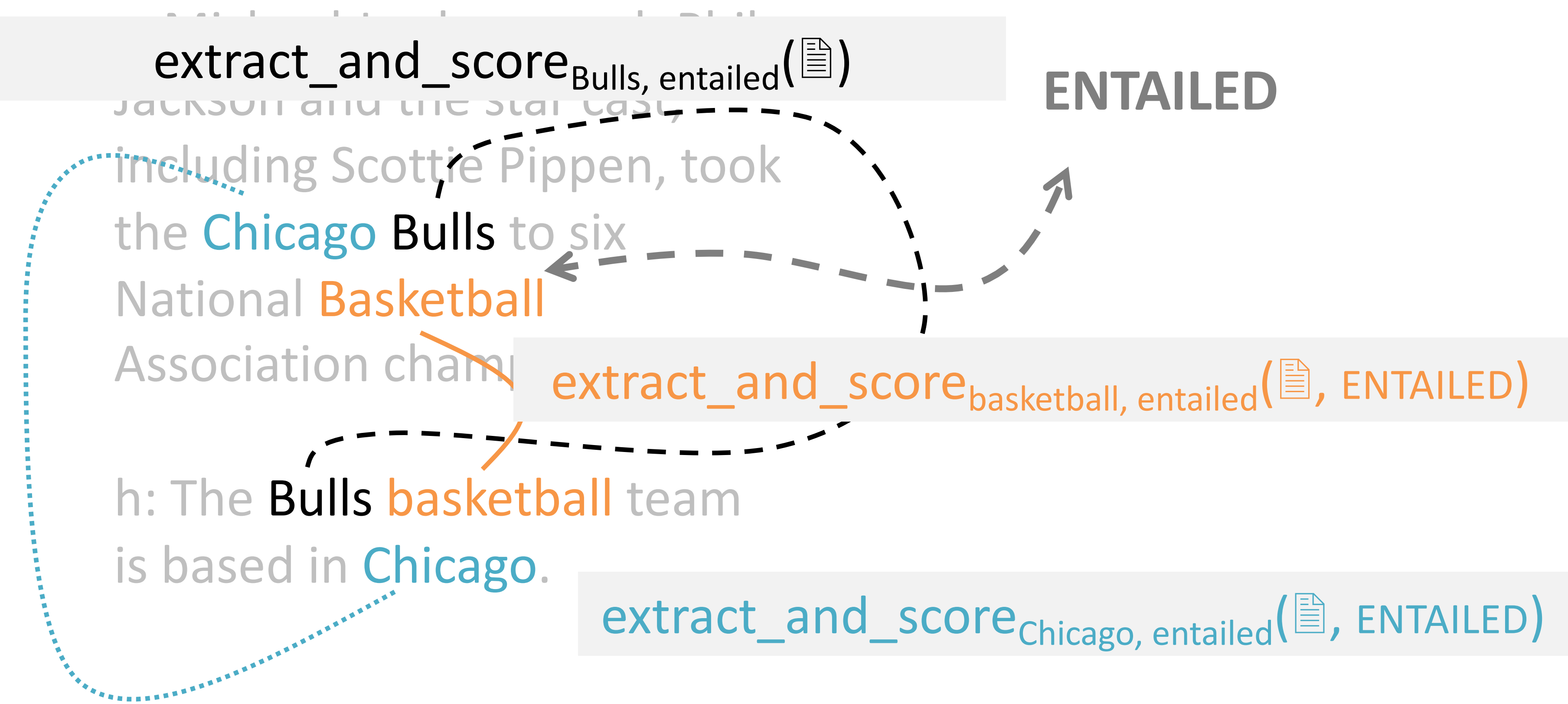
s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.

h: The Bulls basketball team is based in Chicago.

ENTAILED

These extractions are all **features** that have **fired** (likely have some significance)

We need to *score* the different extracted clues.



Michael Jordan and Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association cham

h: The Bulls basketball team is based in Chicago.

$\text{extract\_and\_score}_{\text{Bulls, entailed}}$(🗎)

ENTAILED

$\text{extract\_and\_score}_{\text{basketball, entailed}}$(🗎, ENTAILED)

$\text{extract\_and\_score}_{\text{Chicago, entailed}}$(🗎, ENTAILED)

Score and Combine Our Clues

$\text{score}_{1,\ \text{Entailed}}$(📄)

$\text{score}_{2\ ,\ \text{Entailed}}$(📄)

$\text{score}_{3\ ,\ \text{Entailed}}$(📄)

…

$\text{score}_{k\ ,\ \text{Entailed}}$(📄)

…

COMBINE

→

posterior
probability of
ENTAILED

Scoring Our Clues

score(

s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.

h: The Bulls basketball team is based in Chicago.

, ENTAILED) =

(ignore the feature indexing for now)

$\text{score}_{1\text{ , Entailed}}$(🗎) ✚

$\text{score}_{2\text{ , Entailed}}$(🗎) ✚

$\text{score}_{3\text{ , Entailed}}$(🗎) ✚

…

Turning Scores into Probabilities

score(s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.
h: The Bulls basketball team is based in Chicago. , ENTAILED) > score(s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.
h: The Bulls basketball team is based in Chicago. , NOT ENTAILED)

p(ENTAILED | s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.
h: The Bulls basketball team is based in Chicago.) > p(NOT ENTAILED | s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.
h: The Bulls basketball team is based in Chicago.)

KEY IDEA

Turning Scores into Probabilities (More Generally)

$$\mathrm{score}(x, y_1) > \mathrm{score}(x, y_2)$$

$$\Rightarrow$$

$$p(y_1 | x) > p(y_2 | x)$$

KEY IDEA

Maxent Modeling

$$\underbrace{p\left(\text{ENTAILED} \,\middle|\, \boxed{\begin{array}{l}\text{s: Michael Jordan, coach Phil}\\ \text{Jackson and the star cast,}\\ \text{including Scottie Pippen, took}\\ \text{the Chicago Bulls to six}\\ \text{National Basketball Association}\\ \text{championships.}\\ \text{h: The Bulls basketball team is}\\ \text{based in Chicago.}\end{array}}\right)}_{\textit{This must be a probability}} \propto$$

Convert through function G? What is this function?

$$\overbrace{G}\underbrace{\left(\text{score}\left(\boxed{\begin{array}{l}\text{s: Michael Jordan, coach Phil}\\ \text{Jackson and the star cast,}\\ \text{including Scottie Pippen, took the}\\ \text{Chicago Bulls to six National}\\ \text{Basketball Association}\\ \text{championships.}\\ \text{h: The Bulls basketball team is}\\ \text{based in Chicago.}\end{array}}, \text{ENTAILED}\right)\right)}_{\textit{This could be any real number}}$$

What function G…

operates on any real number?

is never less than 0?

monotonic? (a < b ➔ G(a) < G(b))

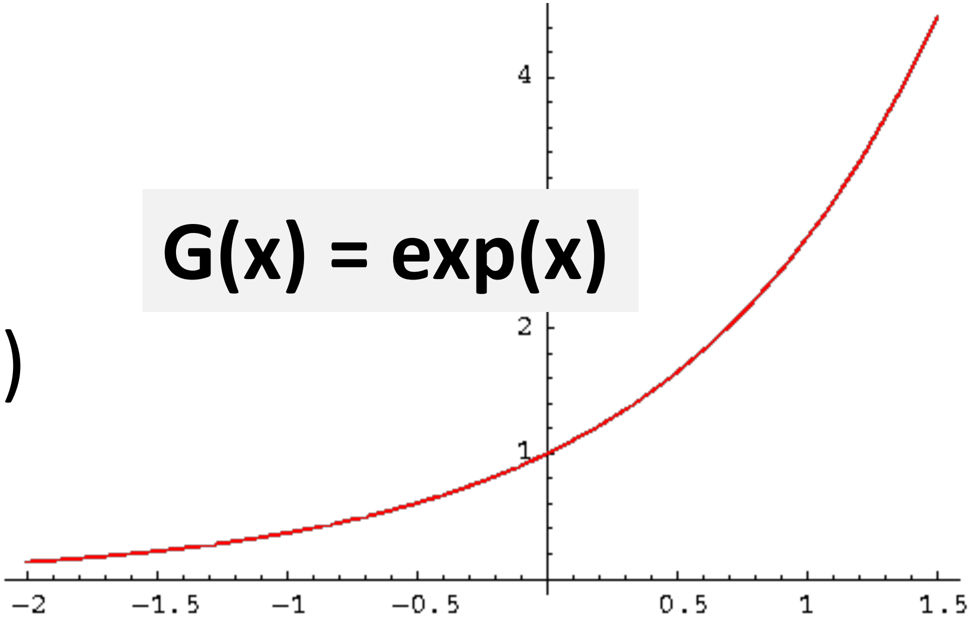
What function G…

operates on any real number?

is never less than 0?

monotonic?

(a < b ➔ G(a) < G(b))



G(x) = exp(x)

Maxent Modeling

$$p\left(\textsc{Entailed} \;\middle|\; \boxed{\begin{array}{l}\text{s: Michael Jordan, coach Phil}\\ \text{Jackson and the star cast,}\\ \text{including Scottie Pippen, took}\\ \text{the Chicago Bulls to six}\\ \text{National Basketball Association}\\ \text{championships.}\\ \text{h: The Bulls basketball team is}\\ \text{based in Chicago.}\end{array}}\right) \propto$$

$$\exp\left(\text{score}\left(\boxed{\begin{array}{l}\text{s: Michael Jordan, coach Phil}\\ \text{Jackson and the star cast, including}\\ \text{Scottie Pippen, took the Chicago}\\ \text{Bulls to six National Basketball}\\ \text{Association championships.}\\ \text{h: The Bulls basketball team is based}\\ \text{in Chicago.}\end{array}}, \textsc{Entailed}\right)\right)$$

Maxent Modeling

$p($ ENTAILED $|$

s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.
h: The Bulls basketball team is based in Chicago.

$) \propto$

$\exp($

$\text{score}_{1\text{ , Entailed}}($🗎$)$ +

$\text{score}_{2\text{ , Entailed}}($🗎$)$ +

$\text{score}_{3\text{ , Entailed}}($🗎$)$ +

…

$))$

Maxent Modeling

$p($ ENTAILED $|$

s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.
h: The Bulls basketball team is based in Chicago.

$) \propto$

exp(

$\text{weight}_{1\text{, Entailed}} * \text{applies}_1($🗎$)$ +

$\text{weight}_{2\text{, Entailed}} * \text{applies}_2($🗎$)$ +

$\text{weight}_{3\text{, Entailed}} * \text{applies}_3($🗎$)$ +

…

))

Maxent Modeling

p(ENTAILED |

s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.

h: The Bulls basketball team is based in Chicago.

) ∝

exp($\text{weight}_{1,\ \text{Entailed}}$ * applies_{1}(📄) +

$\text{weight}_{2,\ \text{Entailed}}$ * applies_{2}(📄) +

$\text{weight}_{3,\ \text{Entailed}}$ * applies_{3}(📄) +

…

))

K different weights…

for K different features

Maxent Modeling

p(ENTAILED |

s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.
h: The Bulls basketball team is based in Chicago.

) $\propto$

exp(
weight$_{1, \text{Entailed}}$ * applies$_1$(🗎) +
weight$_{2, \text{Entailed}}$ * applies$_2$(🗎) +
weight$_{3, \text{Entailed}}$ * applies$_3$(🗎) +
…
))

K different weights…

for K different features

multiplied and then summed

Maxent Modeling

p(ENTAILED |

s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.

h: The Bulls basketball team is based in Chicago.

) ∝

exp(Dot_product of Entailed weight_vec feature_vec(🗎))

K different weights…

for K different features…

multiplied and then summed

Maxent Modeling

$p($ ENTAILED $|$

s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.
h: The Bulls basketball team is based in Chicago.

$) \propto$

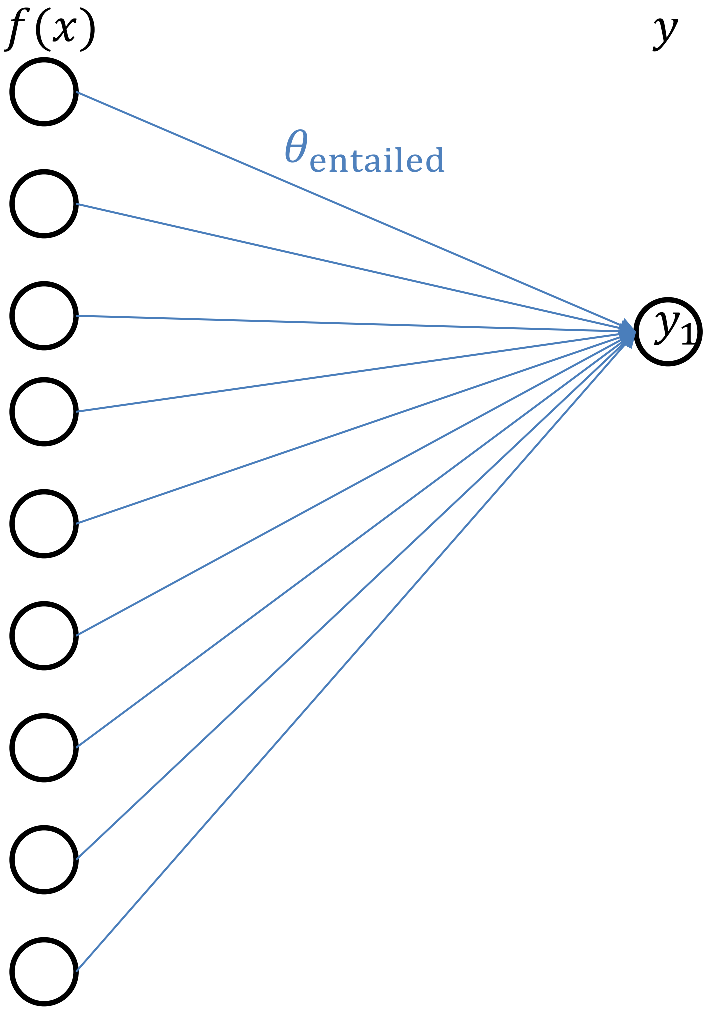
$\exp(\theta^{T}_{\text{ENTAILED}} f($ 📄 $))$

K different weights…

for K different features…

multiplied and then summed

Maxent Classifier, schematically



$p(y = \text{entailed}|x) \propto$
$\exp(\theta_{\text{entailed}} f(x))$

Maxent Modeling

$$p(\text{ENTAILED} \mid \boxed{\begin{array}{l}\text{s: Michael Jordan, coach Phil Jackson and the star cast,}\\ \text{including Scottie Pippen, took the Chicago Bulls to six}\\ \text{National Basketball Association championships.}\\ \text{h: The Bulls basketball team is based in Chicago.}\end{array}}) =$$

Q: How do we define Z?

$$\frac{1}{Z}\exp(\theta_{\text{ENTAILED}}^{T} f(📄))$$

K different weights…

for K different features…

multiplied and then summed

Normalization for Classification

$$Z = \sum_{\text{label } j} \exp\left(\theta_j^T f(\text{📄}) \right)$$

$p(y \mid x) \propto \exp(\theta_y^T f(x))$

classify doc x with label y in one go

Normalization for Classification (long form)

Z =

∑ exp(

label j

weight$_{1, j}$ * applies$_1$(🗎) +

weight$_{2, j}$ * applies$_2$(🗎) +

weight$_{3, j}$ * applies$_3$(🗎) +

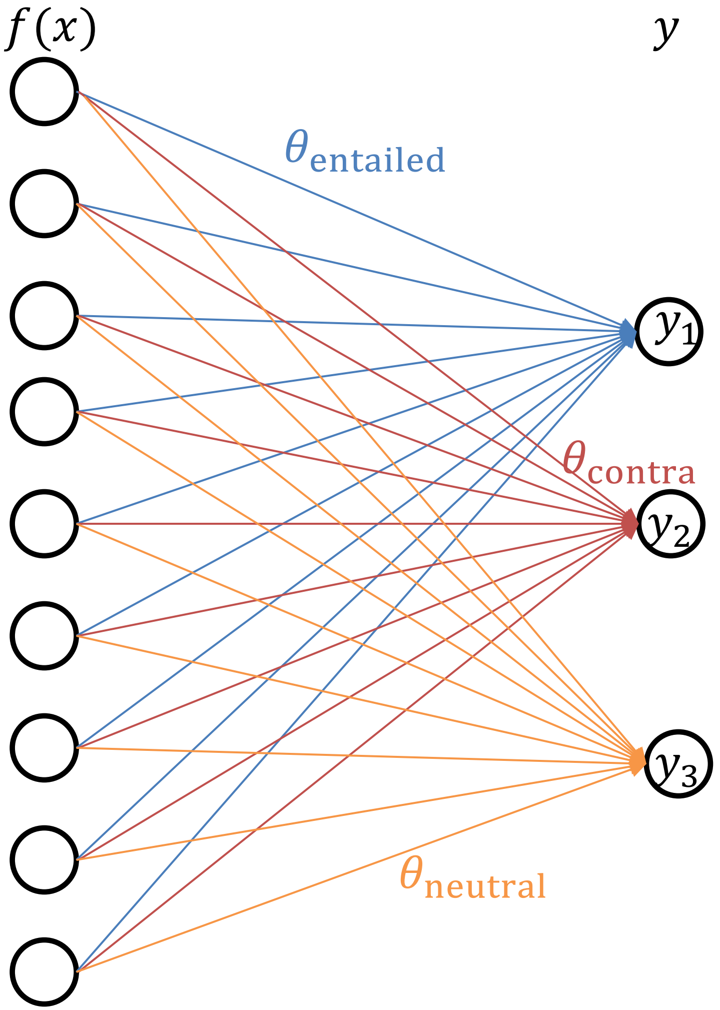
…

)

$p(y \mid x) \propto \exp(\theta_y^T f(x))$

classify doc x with label y in one go

Maxent Classifier, schematically

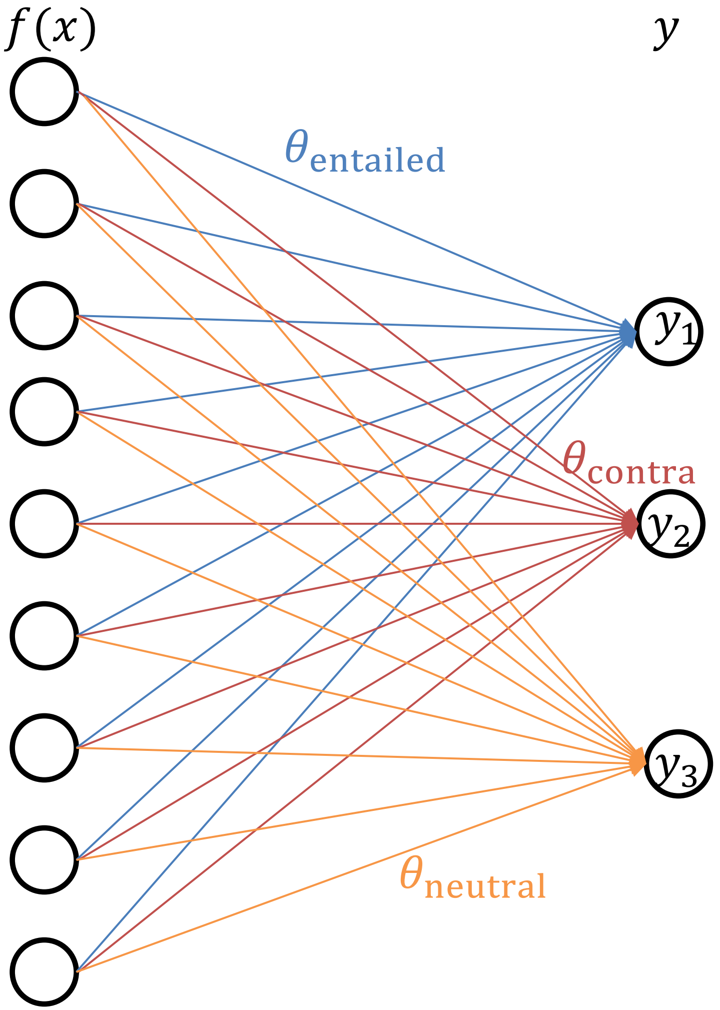


$$p(y = \text{entailed}|x) \propto \exp(\theta_{\text{entailed}} f(x))$$

$$p(y = \text{contra}|x) \propto \exp(\theta_{\text{contra}} f(x))$$

$$p(y = \text{neutral}|x) \propto \exp(\theta_{\text{neutral}} f(x))$$

Maxent Classifier, schematically



$p(y = \text{entailed}|x) \propto$
$\exp(\theta_{\text{entailed}} f(x))$

$p(y = \text{contra}|x) \propto$
$\exp(\theta_{\text{contra}} f(x))$

$p(y = \text{neutral}|x) \propto$
$\exp(\theta_{\text{neutral}} f(x))$

output:
i = argmax score$_i$
class *i*

Core Aspects to Maxent Classifier p(y|x)

- **features** $f(x)$ from x that are meaningful;
- **weights** θ (at least one per feature, often one per feature/label combination) to say how important each feature is; and
- a way to **form probabilities** from f and θ

$$p(y\,|x) = \frac{\exp(\theta_y^T f(x))}{\sum_{y'} \exp(\theta_{y'}^T f(x))}$$

Different Notation, Same Meaning

$$p(Y = y\,|x) = \frac{\exp(\theta_y^T f(x))}{\sum_{y'} \exp(\theta_{y'}^T f(x))}$$

Different Notation, Same Meaning

$$p(Y = y \mid x) = \frac{\exp(\theta_y^T f(x))}{\sum_{y'} \exp(\theta_{y'}^T f(x))}$$

$$p(Y = y \mid x) \propto \exp(\theta_y^T f(x))$$

Different Notation, Same Meaning

$$p(Y = y \mid x) = \frac{\exp(\theta_y^T f(x))}{\sum_{y'} \exp(\theta_{y'}^T f(x))}$$

$$p(Y = y \mid x) \propto \exp(\theta_y^T f(x))$$

$$p(Y \mid x) = \mathrm{softmax}(\theta^T f(x))$$

Outline

Maximum Entropy models

- Defining the model
- Defining the objective
- Learning: Optimizing the objective
- Math: gradient derivation (advanced)

1. Defining Appropriate Features
2. Understanding features in conditional models

Defining Appropriate Features in a Maxent Model

Feature functions help extract useful features (characteristics) of the data

They turn *data* into *numbers*

Features that are not 0 are said to have fired

Generally *templated*

Often binary-valued (0 or 1), but can be real-valued

Reminder!

Bag-of-words as a Function

Based on *some* tokenization, turn an input document into an array (or dictionary or set) of its unique vocab items

Think of getting a BOW rep. as a function **f**

input: Document

output: Container of size *E*, indexable by each vocab type *v*

Some Bag-of-words Functions

Reminder!

Kind	Type of f_v	Interpretation
Binary	0, 1	Did *v* appear in the document?
Count-based	Natural number (int >= 0)	How often did *v* occur in the document?
Averaged	Real number (>=0, <= 1)	How often did *v* occur in the document, normalized by doc length?
TF-IDF (term frequency, inverse document frequency)	Real number (>= 0)	How frequent is a word, tempered by how prevalent it is across the corpus (to be covered later!)
	…	

Q: Is this a reasonable representation?

Q: What are some tradeoffs (benefits vs. costs)?

Templated Features

Define a feature f_{clue}(📄) for each clue you want to consider

The feature f_{clue} fires if the clue applies to/can be found in 📄

Clue is often a target phrase (an n-gram)

Maxent Modeling:
Templated Binary Feature Functions

p(ENTAILED |

s: Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships.
h: The Bulls basketball team is based in Chicago.

) ∝

exp(

weight$_1$, Entailed * applies$_1$(🗎) +

weight$_1$, Entailed * applies$_2$(🗎) +

weight$_1$, Entailed * applies$_3$(🗎) +

…

))

$$\text{applies}_{\text{target}}(🗎) = \begin{cases} 1, \text{target } \textit{matches } 🗎 \\ 0, \qquad \text{otherwise} \end{cases}$$

binary

Example of a Templated Binary Feature Functions

$$\text{applies}_{\text{target}}(\text{🗎}) = \begin{cases} 1, \text{target } \mathit{matches} \text{ 🗎} \\ 0, \qquad \text{otherwise} \end{cases}$$

$$\Downarrow$$

$$\text{applies}_{\text{ball}}(\text{🗎}) = \begin{cases} 1, \text{ball } \mathit{in} \text{ both s and h of 🗎} \\ 0, \qquad \text{otherwise} \end{cases}$$

Example of a Templated Binary Feature Functions

$$\text{applies}_{\text{target}}(🗎) = \begin{cases} 1, \text{target } \mathit{matches}\ 🗎 \\ 0, \qquad \text{otherwise} \end{cases}$$

$$\text{applies}_{\text{ball}}\ (🗎) = \begin{cases} 1, \text{ball } \mathit{in} \text{ both s and h of } 🗎 \\ \qquad\qquad 0, \qquad \text{otherwise} \end{cases}$$

Q: If there are V vocab types and L label types:

1. How many features are defined if unigram targets are used (w/ each label)?

Example of a Templated Binary Feature Functions

$$\text{applies}_{\text{target}}(\text{🗎}) = \begin{cases} 1, \text{target } \mathit{matches} \text{ 🗎} \\ 0, \quad\quad \text{otherwise} \end{cases}$$

⇓

$$\text{applies}_{\text{ball}}\ (\text{🗎}) = \begin{cases} 1, \text{ball } \mathit{in} \text{ both s and h of 🗎} \\ \quad\quad\quad\quad\quad\quad 0, \quad\quad \text{otherwise} \end{cases}$$

Q: If there are V vocab types and L label types:

1. How many features are defined if unigram targets are used (w/ each label)?

A1: VL

Example of a Templated Binary Feature Functions

$$\text{applies}_{\text{target}}(\text{🗎}) = \begin{cases} 1, \text{target } \textit{matches} \text{ 🗎} \\ 0, \quad\quad \text{otherwise} \end{cases}$$

$$\text{applies}_{\text{ball}}(\text{🗎}) = \begin{cases} 1, \text{ball } \textit{in} \text{ both s and h of 🗎} \\ 0, \quad\quad \text{otherwise} \end{cases}$$

Q: If there are V vocab types and L label types:

1. How many features are defined if unigram targets are used (w/ each label)?

A1: VL

2. How many features are defined if bigram targets are used?

Example of a Templated Binary Feature Functions

$$\text{applies}_{\text{target}}(🗎) = \begin{cases} 1, \text{target } \mathit{matches}\ 🗎 \\ 0, \quad \text{otherwise} \end{cases}$$

$$\text{applies}_{\text{ball}}\ (🗎) = \begin{cases} 1, \text{ball } \mathit{in} \text{ both s and h of } 🗎 \\ 0, \quad \text{otherwise} \end{cases}$$

Q: If there are V vocab types and L label types:

1. How many features are defined if unigram targets are used (w/ each label)?

A1: VL

2. How many features are defined if bigram targets are used (w/ each label)?

A2: V^2L

Example of a Templated Binary Feature Functions

$$\text{applies}_{\text{target}}(🗎) = \begin{cases} 1, \text{target } \mathit{matches}\ 🗎 \\ 0, \quad \text{otherwise} \end{cases}$$

$$\text{applies}_{\text{ball}}\ (🗎) = \begin{cases} 1, \text{ball } \mathit{in} \text{ both s and h of } 🗎 \\ \quad 0, \quad \text{otherwise} \end{cases}$$

Q: If there are V vocab types and L label types:

1. How many features are defined if unigram targets are used (w/ each label)?

A1: VL

2. How many features are defined if bigram targets are used (w/ each label)?

A2: V^2L

3. How many features are defined if unigram and bigram targets are used (w/ each label)?

Example of a Templated Binary Feature Functions

$$\text{applies}_{\text{target}}(\text{🗎}) = \begin{cases} 1, \text{target } \mathit{matches}\ \text{🗎} \\ 0, \quad \text{otherwise} \end{cases}$$

↓

$$\text{applies}_{\text{ball}}\ (\text{🗎}) = \begin{cases} 1, \text{ball } \mathit{in} \text{ both s and h of } \text{🗎} \\ 0, \quad \text{otherwise} \end{cases}$$

Q: If there are V vocab types and L label types:

1. How many features are defined if unigram targets are used (w/ each label)?

A1: VL

2. How many features are defined if bigram targets are used (w/ each label)?

A2: V^2L

3. How many features are defined if unigram and bigram targets are used (w/ each label)?

A2: $(\mathrm{V} + V^2)L$

Outline

Maximum Entropy models

Defining the model

Defining the objective

Learning: Optimizing the objective

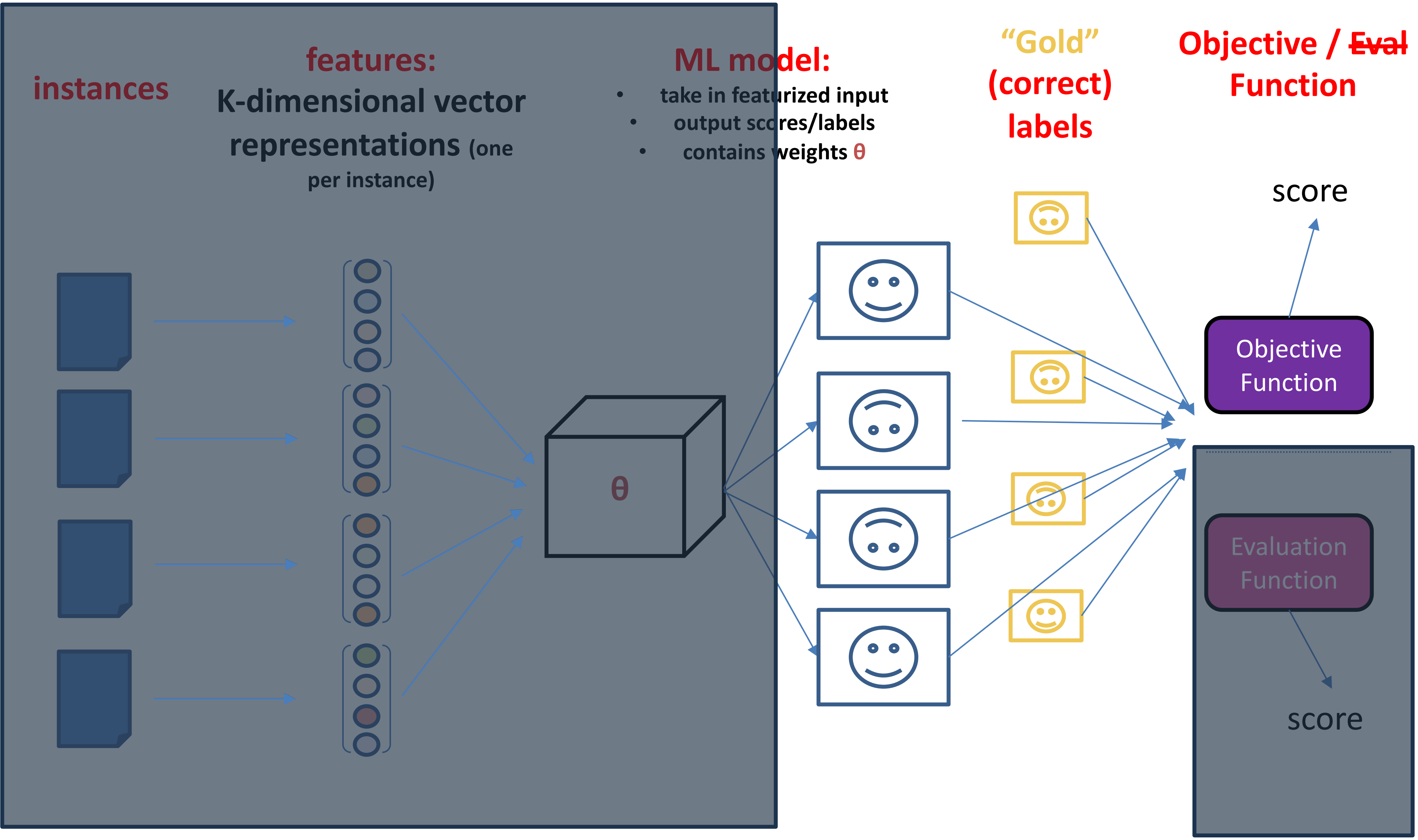
Math: gradient derivation (advanced)

$$p_{\theta}(y \mid x)$$ probabilistic model

$$\Downarrow$$

$$F(\theta; x, y)$$ **objective**

Defining the Objective



Primary Objective: Likelihood

- Goal: *maximize* the score your model gives to the training data it observes

- This is called the **likelihood of your data**

- In classification, this is p(label | 🗎)
- For language modeling, this is p(🗎 | label)

Objective = Full Likelihood? (Classification)

$$\prod_i p_\theta(y_i|x_i) \propto \prod_i \exp(\theta_{y_i}^T f(x_i))$$

These values can have very small magnitude ➔ underflow

Differentiating this product could be a pain

Logarithms

(0, 1] ➔ (-∞, 0]

Products ➔ Sums

$$\log(ab) = \log(a) + \log(b)$$

$$\log(a/b) = \log(a) – \log(b)$$

Inverse of exp

$$\log(\exp(x)) = x$$

Log-Likelihood (Classification)

Wide range of (negative) numbers

Sums are more stable

$$\log \prod_i p_\theta(y_i|x_i) = \sum_i \log p_\theta(y_i|x_i)$$

Products ➔ Sums

log(ab) = log(a) + log(b)

log(a/b) = log(a) – log(b)

Maximize Log-Likelihood (Classification)

Wide range of (negative) numbers

Sums are more stable

$$\log \prod_i p_\theta(y_i|x_i) = \sum_i \log p_\theta(y_i|x_i)$$

Inverse of exp

log(exp(x)) = x

$$= \sum_i \underbrace{\theta_{y_i}^T f(x_i) - \log Z(x_i)}_{\text{Differentiating this becomes nicer (even though Z depends on θ)}}$$

Log-Likelihood (Classification)

Wide range of (negative) numbers

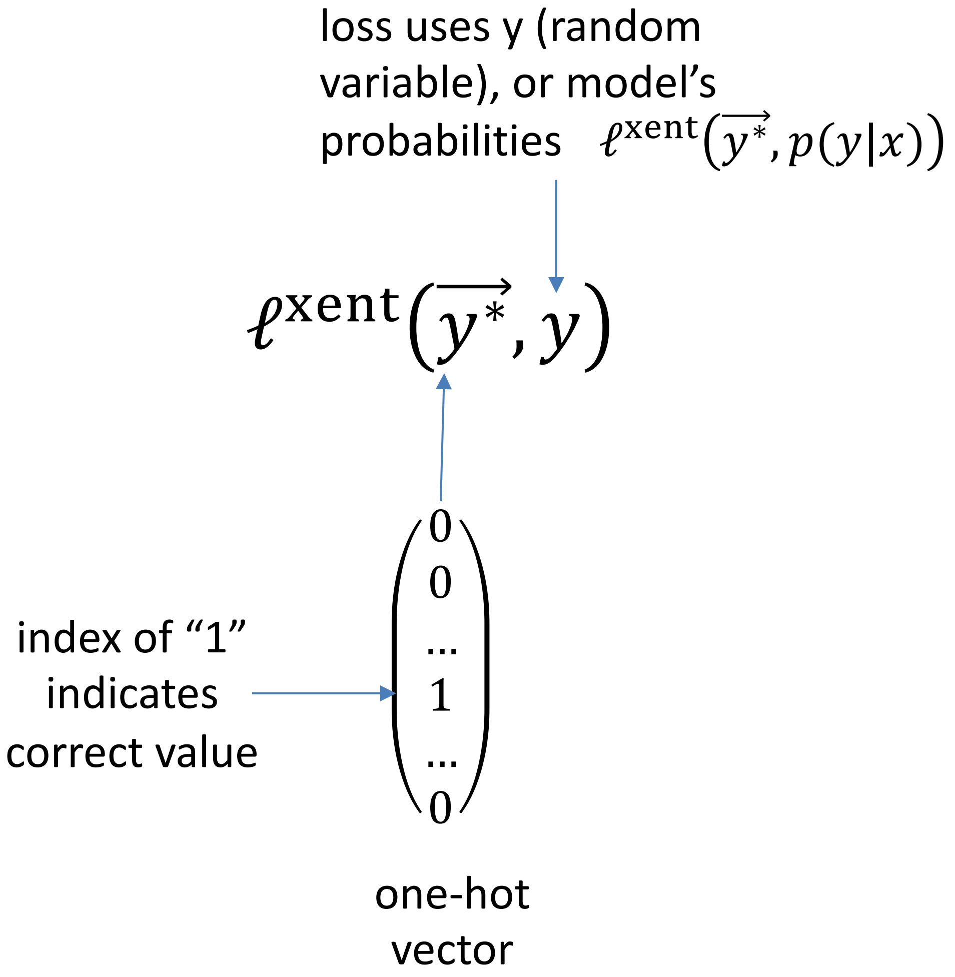
Sums are more stable

$$\log \prod_i p_\theta(y_i|x_i) = \sum_i \log p_\theta(y_i|x_i)$$

$$= \sum_i \theta_{y_i}^T f(x_i) - \log Z(x_i)$$

$$= F(\theta)$$

Equivalent Version 2:
Minimize Cross Entropy Loss



loss uses y (random variable), or model's probabilities $\ell^{\text{xent}}\left(\overrightarrow{y^*}, p(y|x)\right)$

$$\ell^{\text{xent}}\left(\overrightarrow{y^*}, y\right)$$

$$\begin{pmatrix} 0 \\ 0 \\ \ldots \\ 1 \\ \ldots \\ 0 \end{pmatrix}$$

index of "1" indicates correct value

one-hot vector

Equivalent Version 2: *Minimize* Cross Entropy Loss

loss uses y (random variable), or model's probabilities $\ell^{\text{xent}}\left(\overrightarrow{y^*}, p(y|x)\right)$

$$\ell^{\text{xent}}\left(\overrightarrow{y^*}, y\right) = -\sum_k \overrightarrow{y^*}[k] \log p(y = k|x)$$

index of "1" indicates correct value

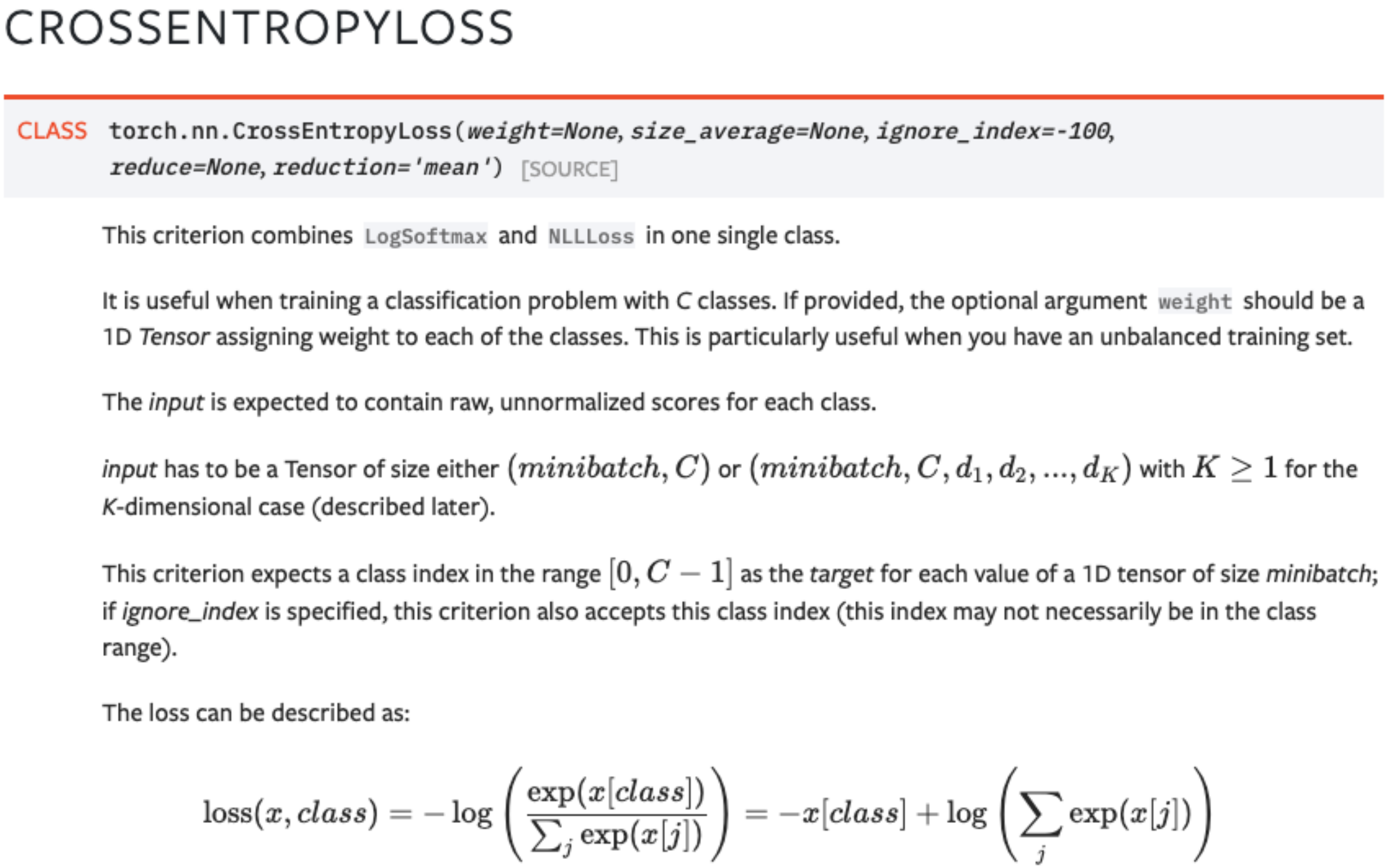
$$\begin{pmatrix} 0 \\ 0 \\ \dots \\ 1 \\ \dots \\ 0 \end{pmatrix}$$

one-hot vector

- minimize xent loss ↔ maximize log-likelihood
- objective is convex

Classification Log-likelihood ≅ Cross Entropy Loss

$$F(\theta) = \sum_i \theta_{y_i}^T f(x_i) - \log Z(x_i)$$



CROSSENTROPYLOSS

CLASS `torch.nn.CrossEntropyLoss(weight=None, size_average=None, ignore_index=-100, reduce=None, reduction='mean')` [SOURCE]

This criterion combines `LogSoftmax` and `NLLLoss` in one single class.

It is useful when training a classification problem with C classes. If provided, the optional argument `weight` should be a 1D *Tensor* assigning weight to each of the classes. This is particularly useful when you have an unbalanced training set.

The *input* is expected to contain raw, unnormalized scores for each class.

input has to be a Tensor of size either $(minibatch, C)$ or $(minibatch, C, d_1, d_2, ..., d_K)$ with $K \geq 1$ for the *K*-dimensional case (described later).

This criterion expects a class index in the range $[0, C-1]$ as the *target* for each value of a 1D tensor of size *minibatch*; if *ignore_index* is specified, this criterion also accepts this class index (this index may not necessarily be in the class range).

The loss can be described as:

$$\text{loss}(x, class) = -\log\left(\frac{\exp(x[class])}{\sum_j \exp(x[j])}\right) = -x[class] + \log\left(\sum_j \exp(x[j])\right)$$

Preventing Extreme Values

- Likelihood on its own can lead to overfitting and/or extreme values in the probability computation

$$F(\theta) = \sum_i \theta_{y_i}^T f(x_i) - \log Z(x_i)$$

Learn the parameters based on
some (fixed) data/examples

Regularization: Preventing Extreme Values

$$F(\theta) = \sum_i \theta_{y_i}^T f(x_i) - \log Z(x_i)$$

With fixed/predefined features, the values of θ determine how "good" or "bad" our objective learning is

Regularization: Preventing Extreme Values

$$F(\theta) = \left(\sum_i \theta_{y_i}^T f(x_i) - \log Z(x_i)\right) - R(\theta)$$

- Augment the objective with a **regularizer**
- This regularizer places an inductive bias (or, prior) on the general "shape" and values of θ

With fixed/predefined features, the values of θ determine how "good" or "bad" our objective learning is

(Squared) L2 Regularization

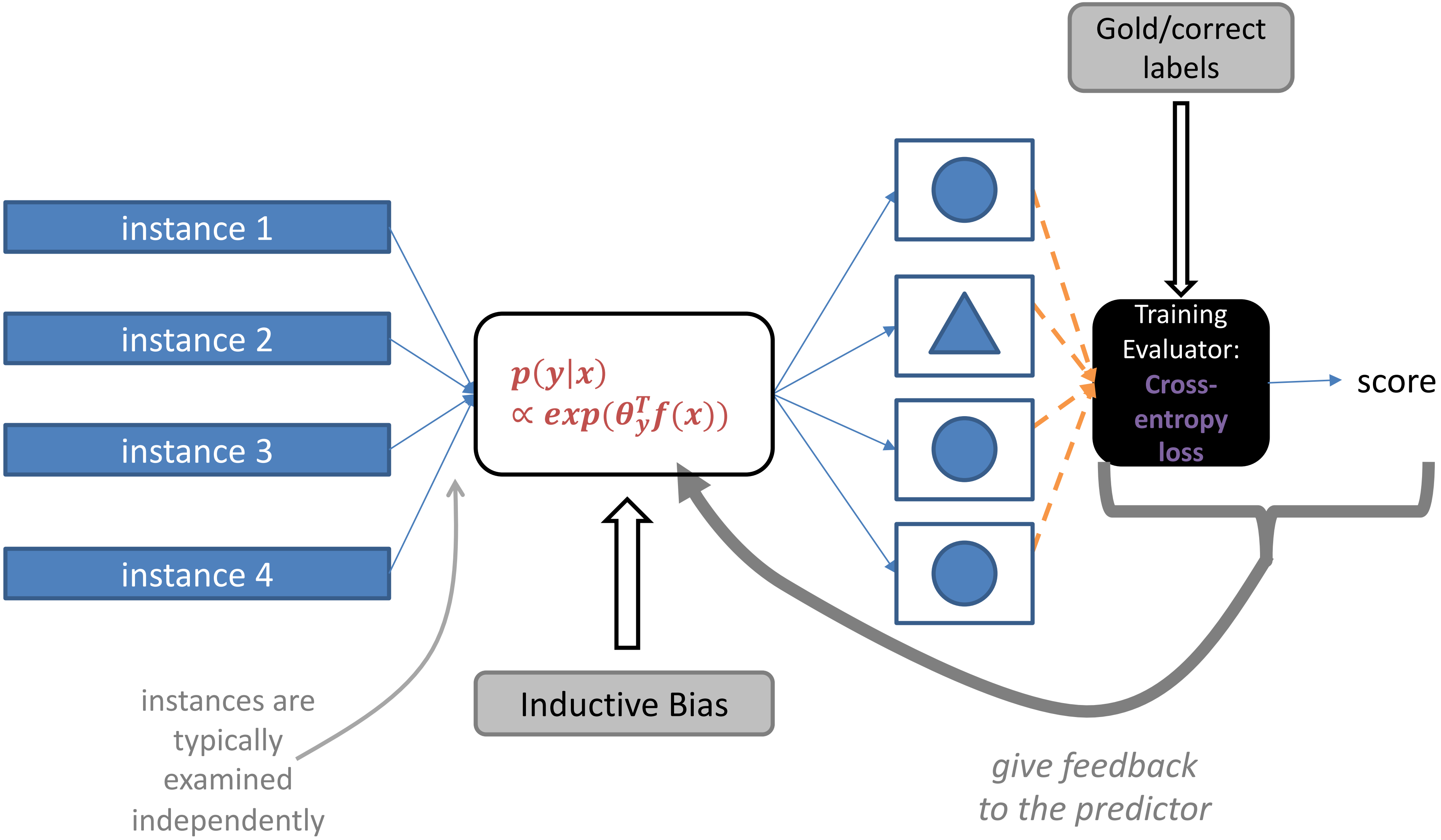
$$R(\theta) = \|\theta\|_2^2 = \sum_k \theta_k^2$$

Outline

Maximum Entropy classifiers

- Defining the model
- Defining the objective
- Learning: Optimizing the objective
- Math: gradient derivation (advanced)

How do we learn?



instance 1

instance 2

instance 3

instance 4

$p(y|x)$
$\propto exp(\theta_y^T f(x))$

Inductive Bias

Gold/correct labels

Training Evaluator: Cross-entropy loss

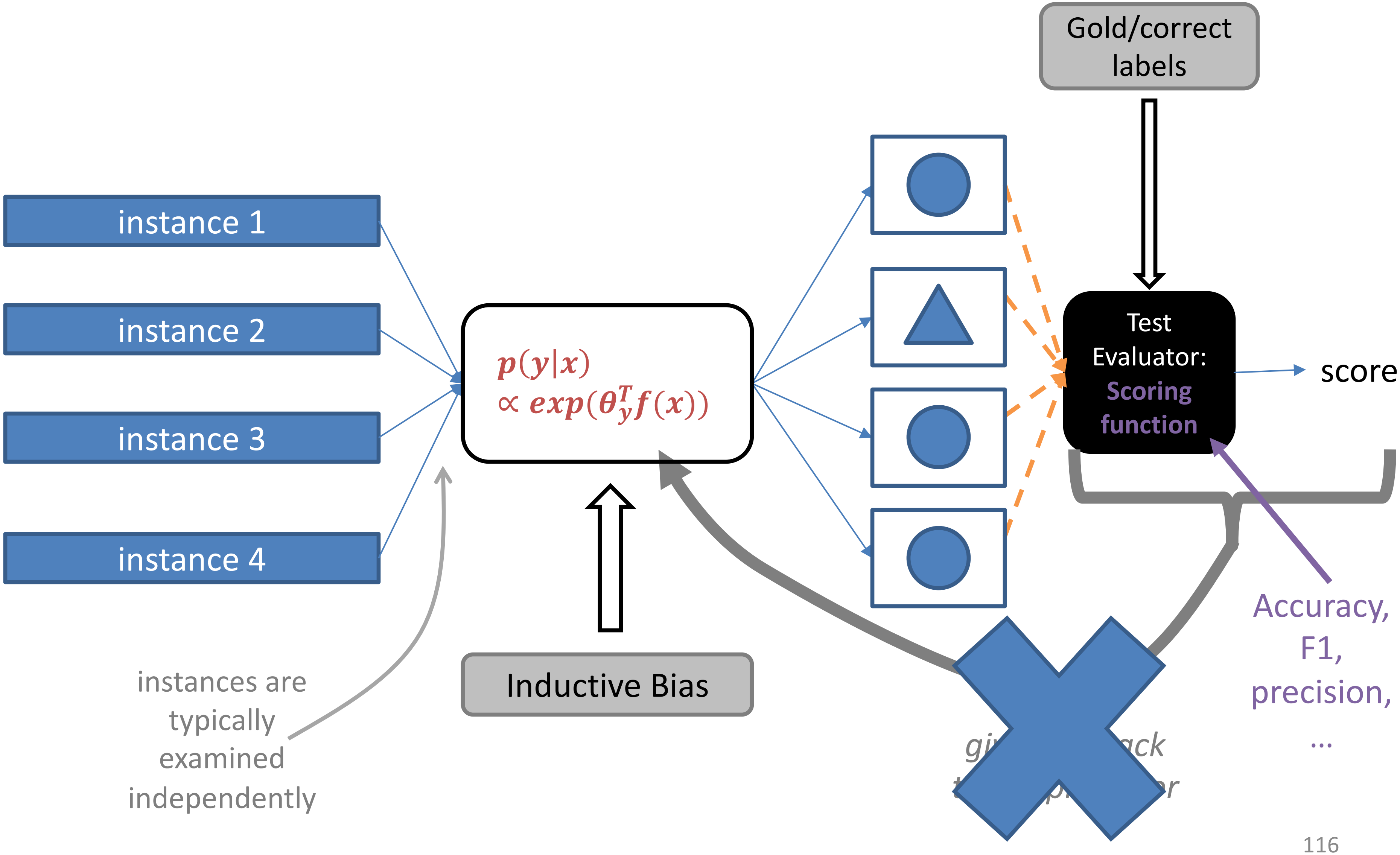
score

instances are typically examined independently

give feedback to the predictor

How do we evaluate (or use)?

Change the eval function.



instance 1

instance 2

instance 3

instance 4

$p(y|x)$
$\propto exp(\theta_y^T f(x))$

Inductive Bias

Gold/correct labels

Test Evaluator: **Scoring function**

score

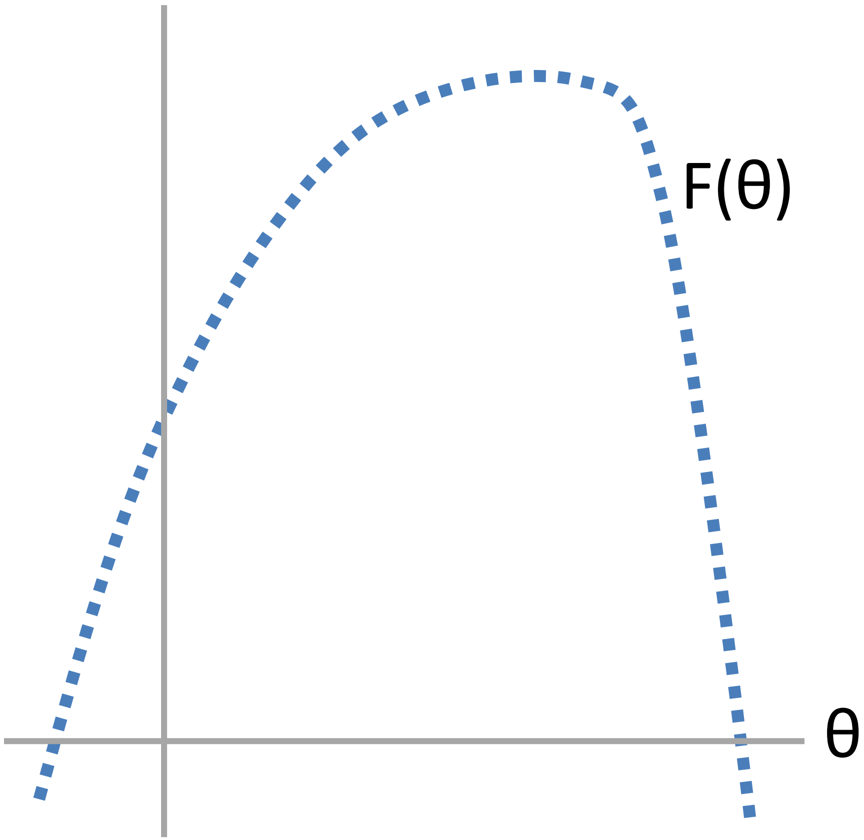
Accuracy, F1, precision, …

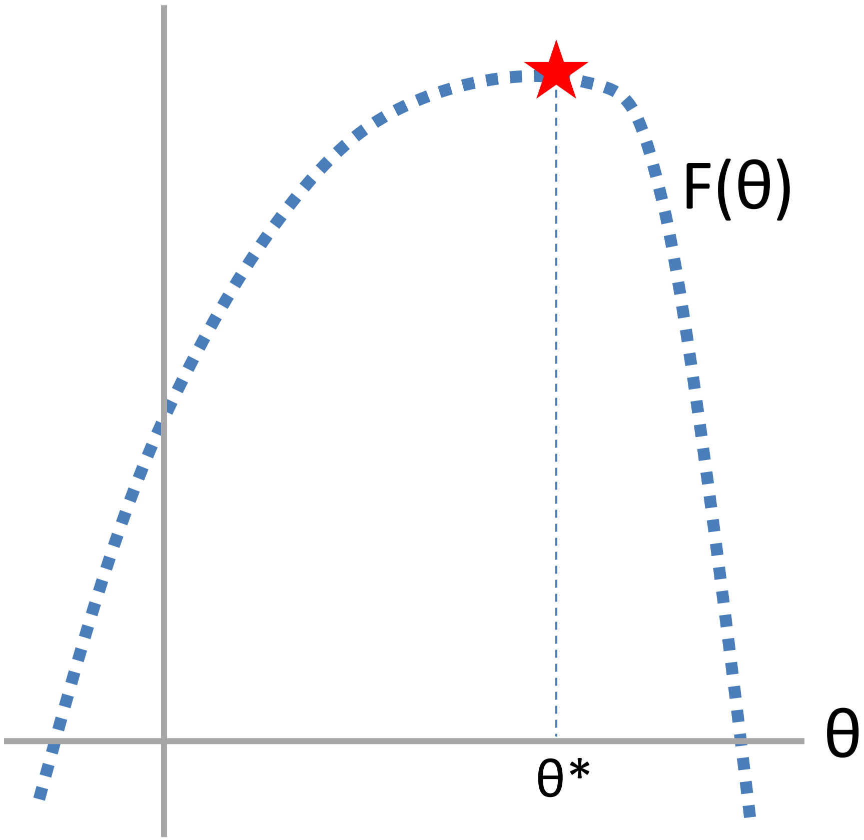
instances are typically examined independently

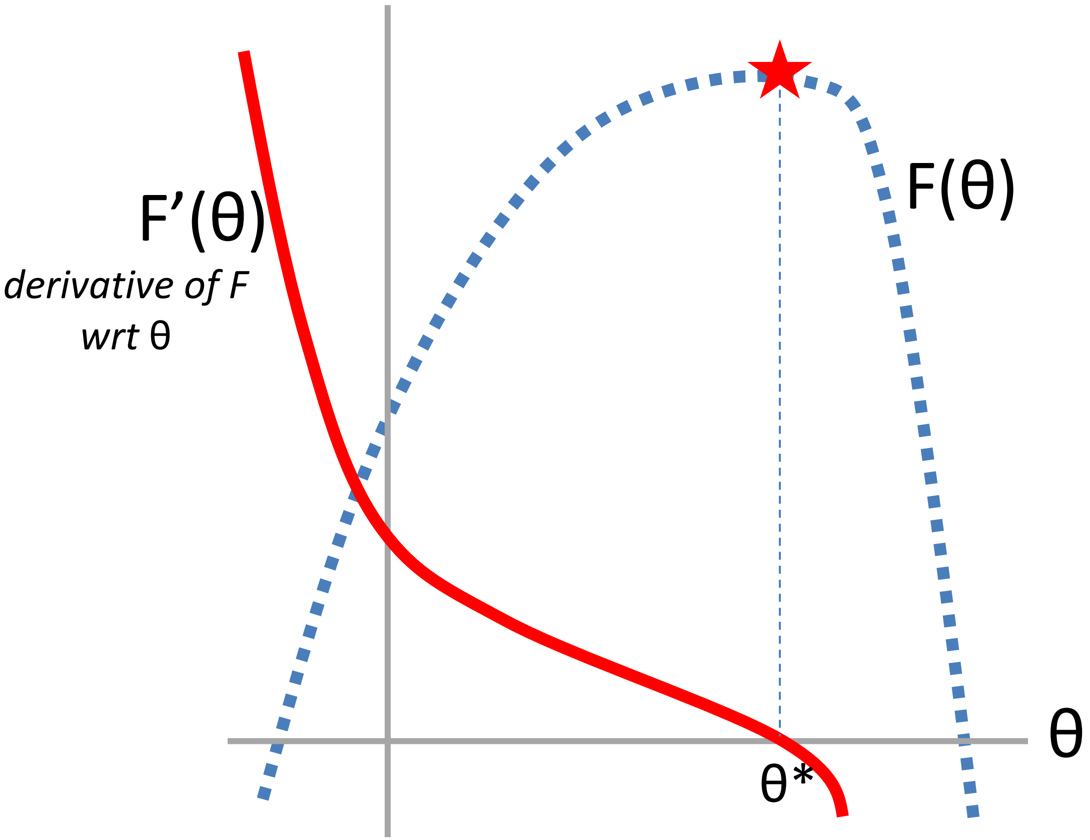
gi ack
t pi r

How will we optimize F(θ)?

Calculus


F(θ)
θ

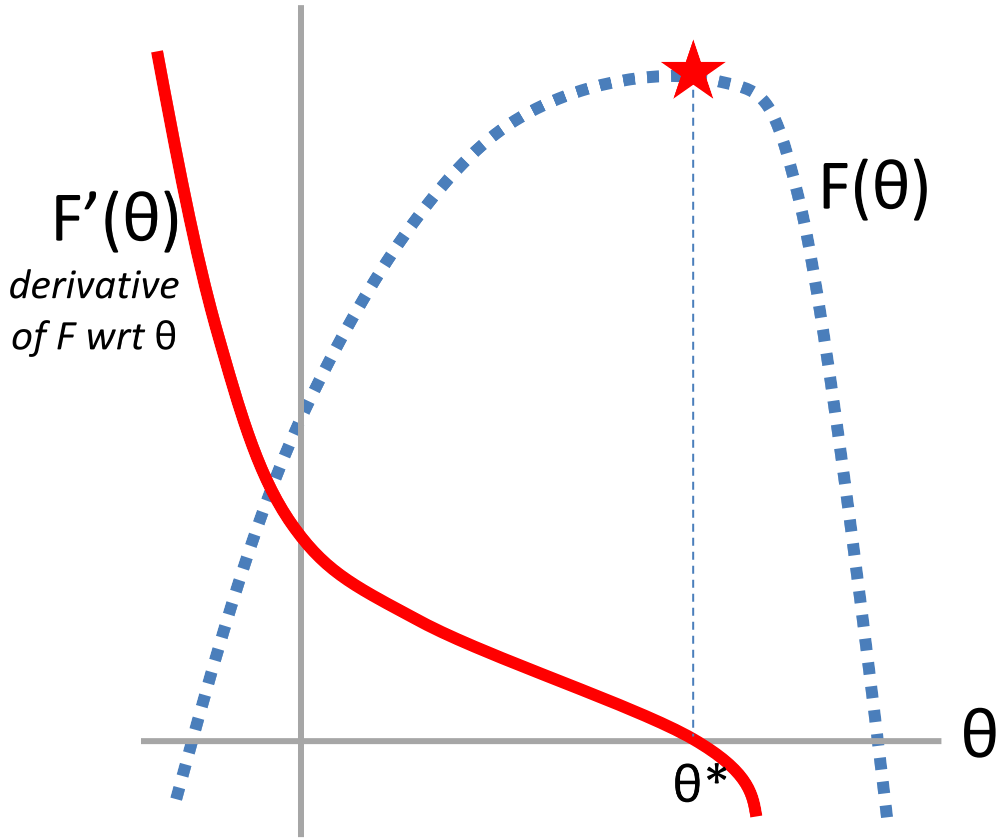

F(θ)
θ
θ*


F’(θ)
derivative of F
wrt θ
F(θ)
θ
θ*

Example (Best case, solve for roots of the derivative)

$$\begin{tikzcd}
F(x) = -(x-2)^2 \arrow[d, "\textit{differentiate}"] \\
F'(x) = -2x + 4 \arrow[d, "\textit{Solve } F'(x) = 0"] \\
x = 2
\end{tikzcd}$$

What if you can’t find the roots? Follow the derivative



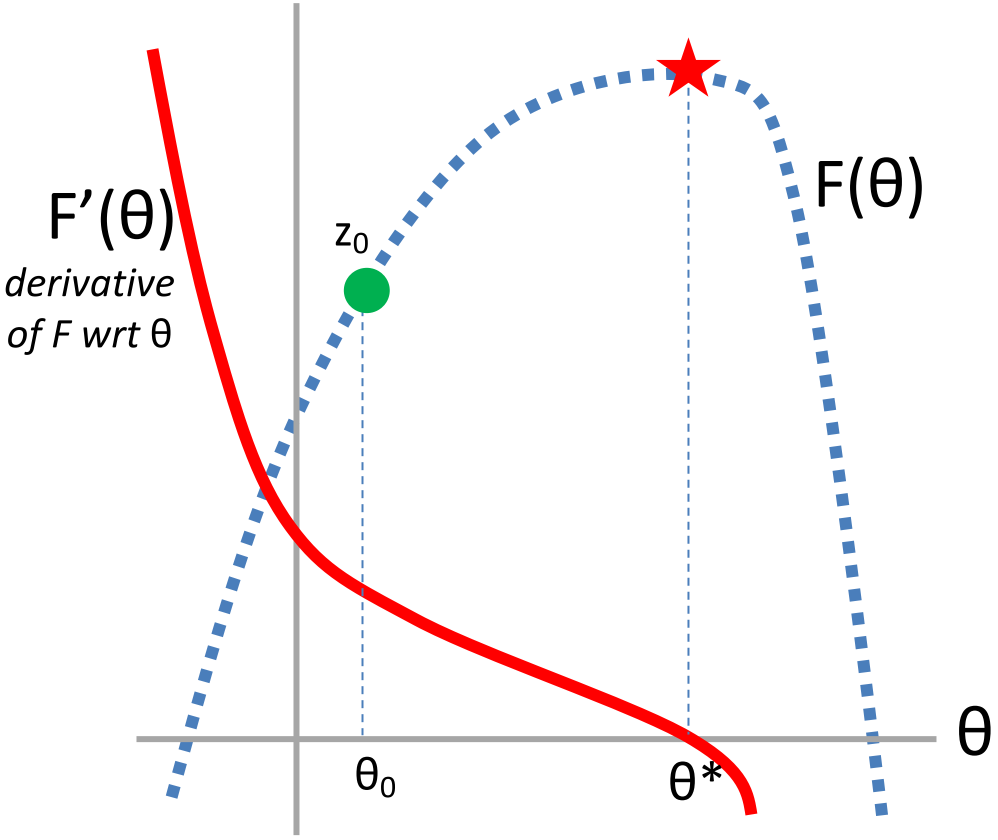
What if you can't find the roots? Follow the derivative

Set t = 0

Pick a starting value θ_t

Until converged:

1. Get value $z_t = F(\theta_t)$



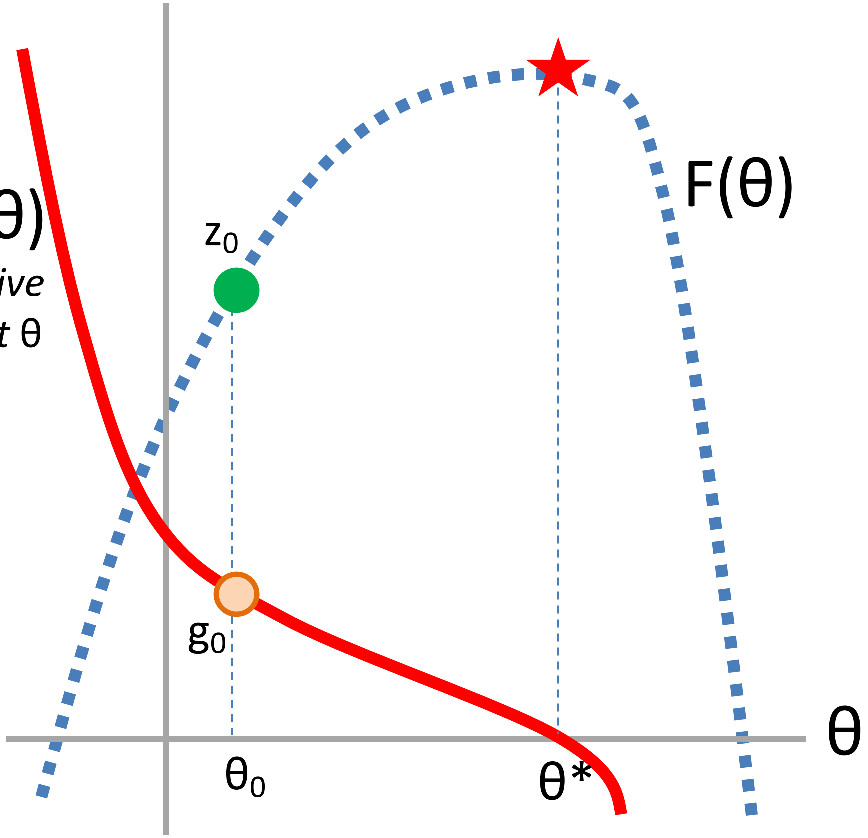
What if you can't find the roots? Follow the derivative

Set t = 0

Pick a starting value θ_t

Until converged:

1. Get value $z_t = F(\theta_t)$
2. Get derivative $g_t = F'(\theta_t)$



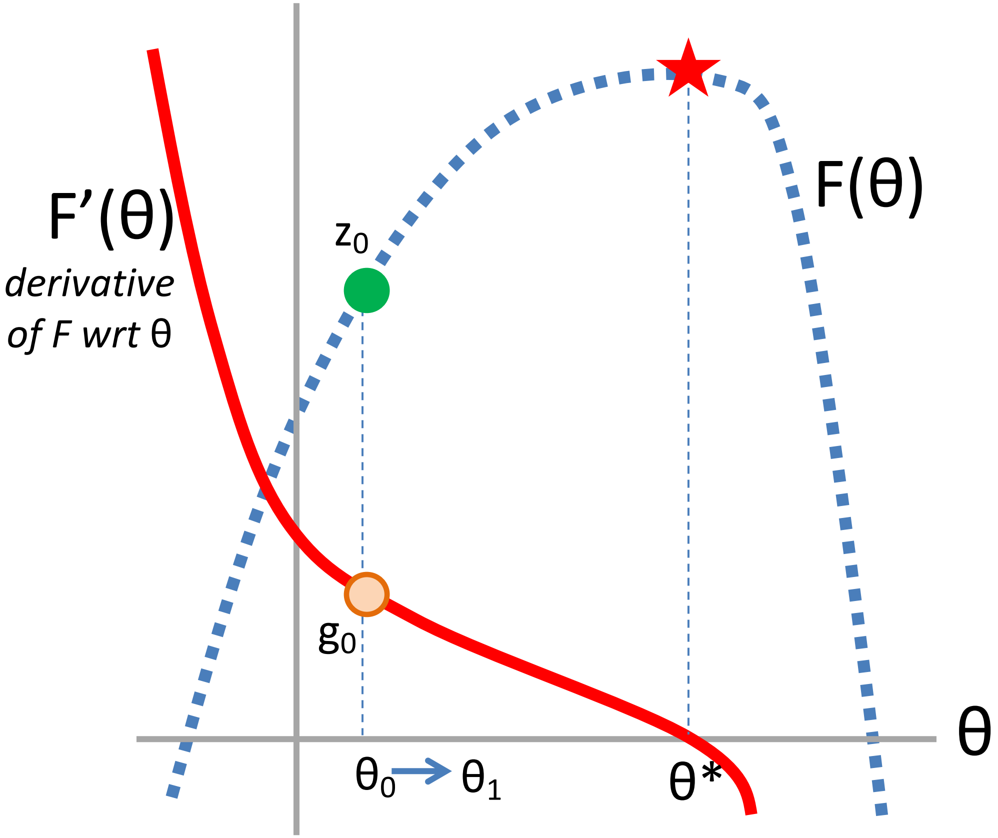
What if you can’t find the roots? Follow the derivative

Set t = 0

Pick a starting value θ_t

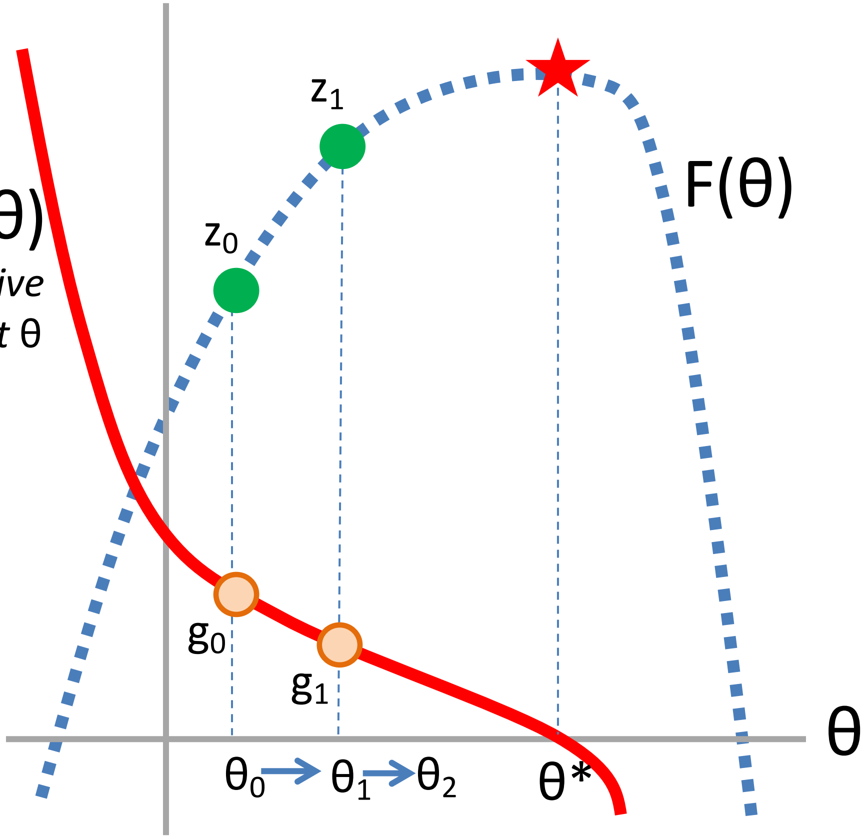
Until converged:

1. Get value $z_t = F(\theta_t)$
2. Get derivative $g_t = F'(\theta_t)$
3. Get scaling factor ρ_t
4. Set $\theta_{t+1} = \theta_t + \rho_t * g_t$
5. Set t += 1



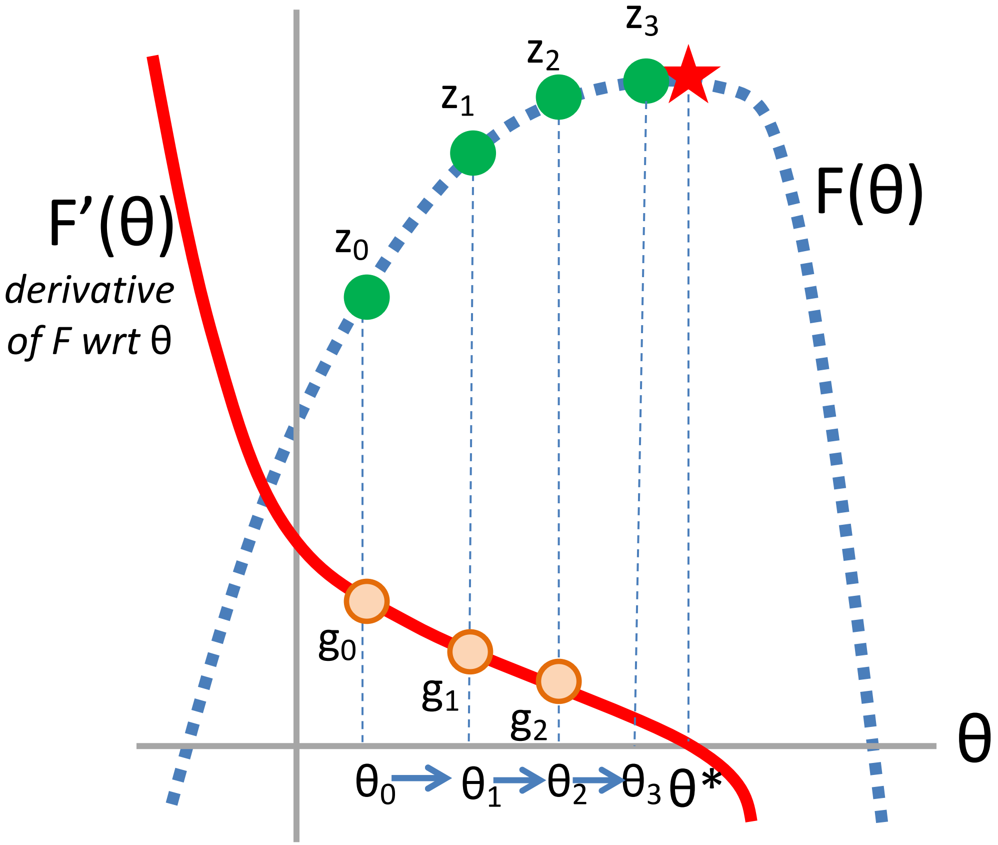
What if you can’t find the roots? Follow the derivative

```
Set t = 0
Pick a starting value $\theta_t$
Until converged:
    1. Get value $z_t = F(\theta_t)$
    2. Get derivative $g_t = F'(\theta_t)$
    3. Get scaling factor $\rho_t$
    4. Set $\theta_{t+1} = \theta_t + \rho_t * g_t$
    5. Set t += 1
```



What if you can’t find the roots? Follow the derivative

```
Set $t = 0$
Pick a starting value $\theta_t$
Until converged:
    1. Get value $z_t = F(\theta_t)$
    2. Get derivative $g_t = F'(\theta_t)$
    3. Get scaling factor $\rho_t$
    4. Set $\theta_{t+1} = \theta_t + \rho_t * g_t$
    5. Set $t$ += 1
```



z_3
z_2
z_1
z_0
F’(θ)
derivative of F wrt θ
F(θ)
g_0
g_1
g_2
θ
$\theta_0 \rightarrow \theta_1 \rightarrow \theta_2 \rightarrow \theta_3 \; \theta^*$

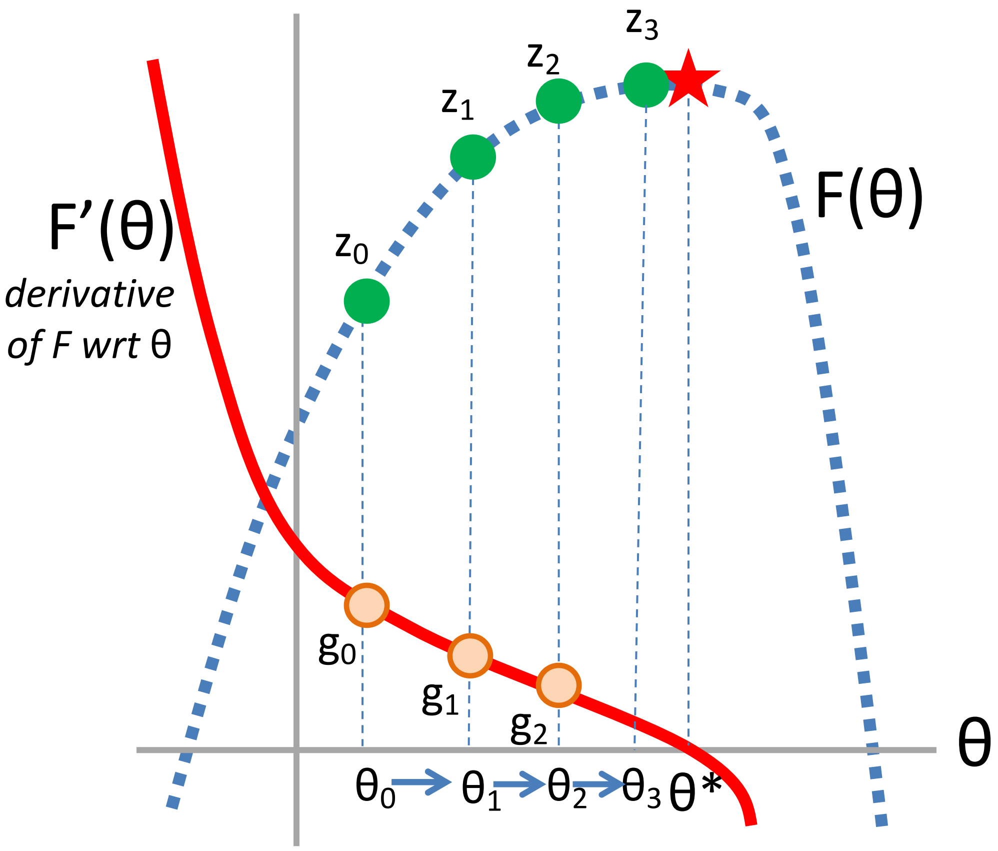
What if you can’t find the roots? Follow the derivative

Set t = 0

Pick a starting value θ_t

Until **converged**:

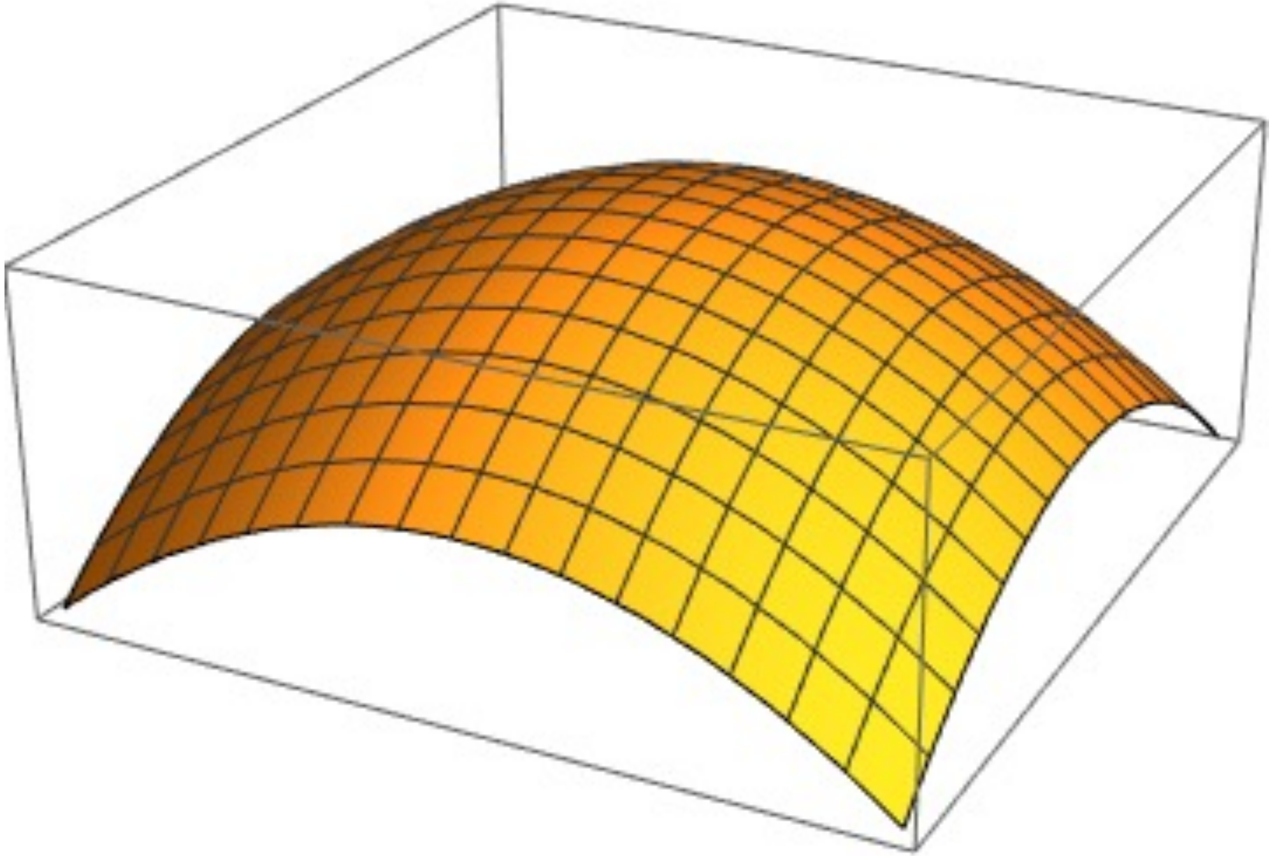
1. Get value $z_t = F(\theta_t)$
2. Get derivative $g_t = F'(\theta_t)$
3. Get **scaling factor ρ_t**
4. Set $\theta_{t+1} = \theta_t + \rho_t * g_t$
5. Set t += 1



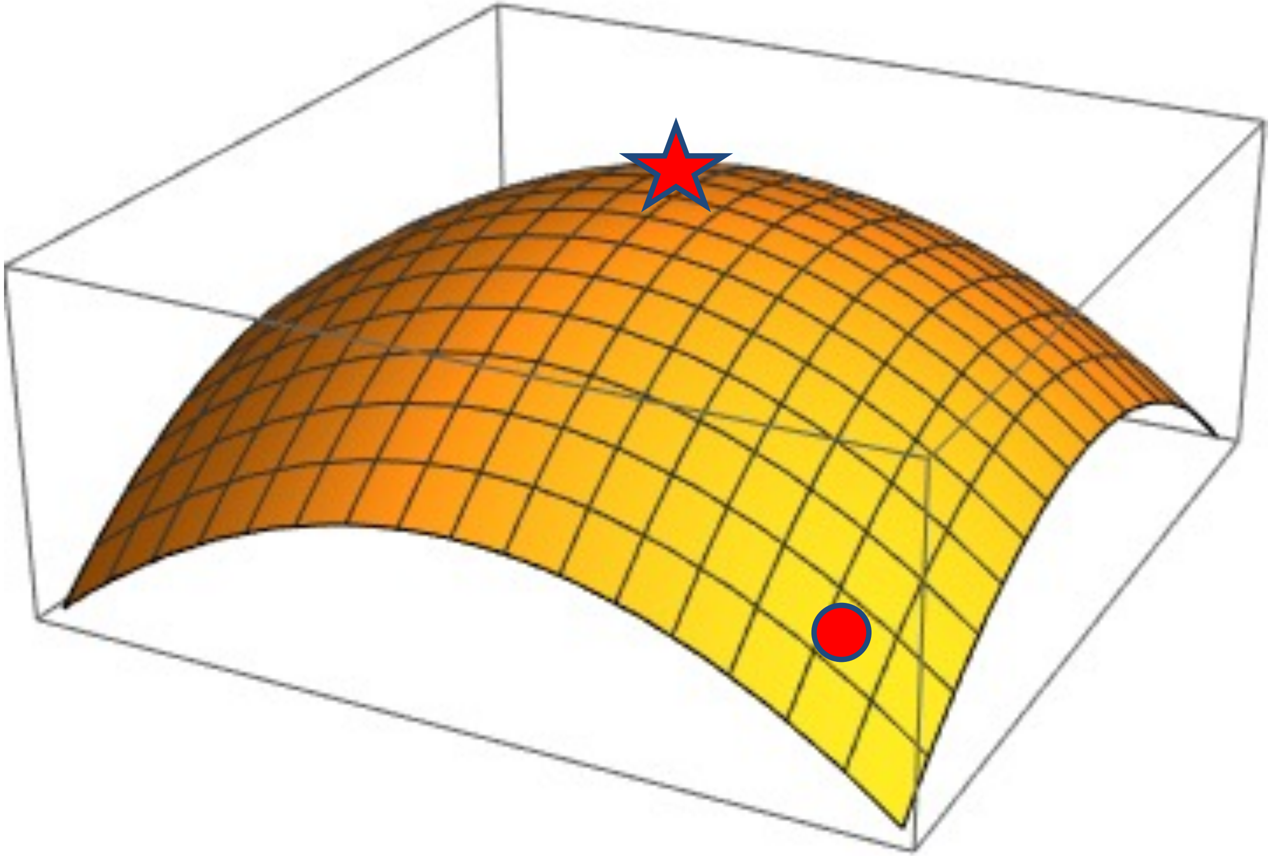
Gradient = Multi-variable derivative

$$\nabla_\theta F\left(\overset{\overset{\text{K-dimensional input}}{\downarrow}}{\theta}\right) = \left(\underbrace{\frac{\partial F}{\partial \theta_1}, \frac{\partial F}{\partial \theta_2}, \ldots, \frac{\partial F}{\partial \theta_K}}_{\text{K-dimensional output}}\right)$$

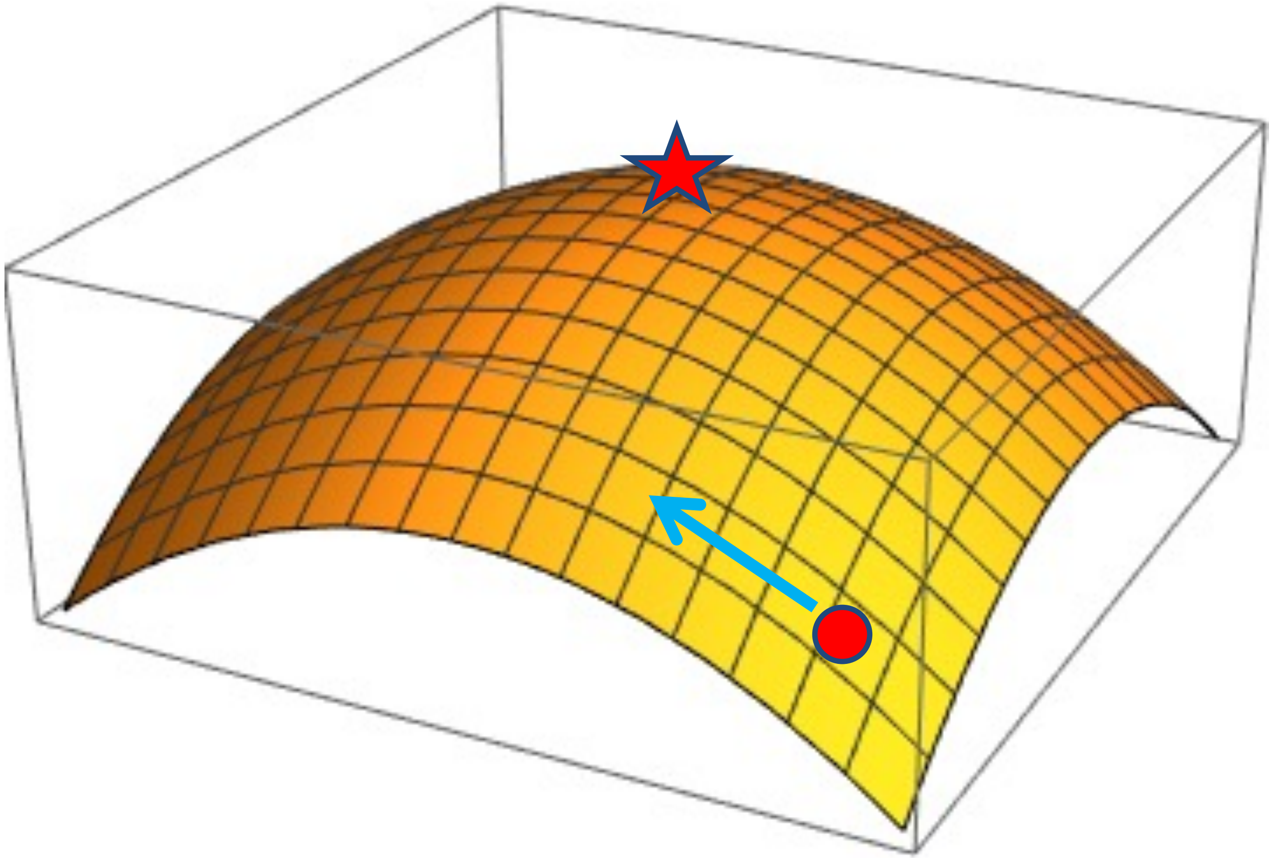
Gradient Ascent



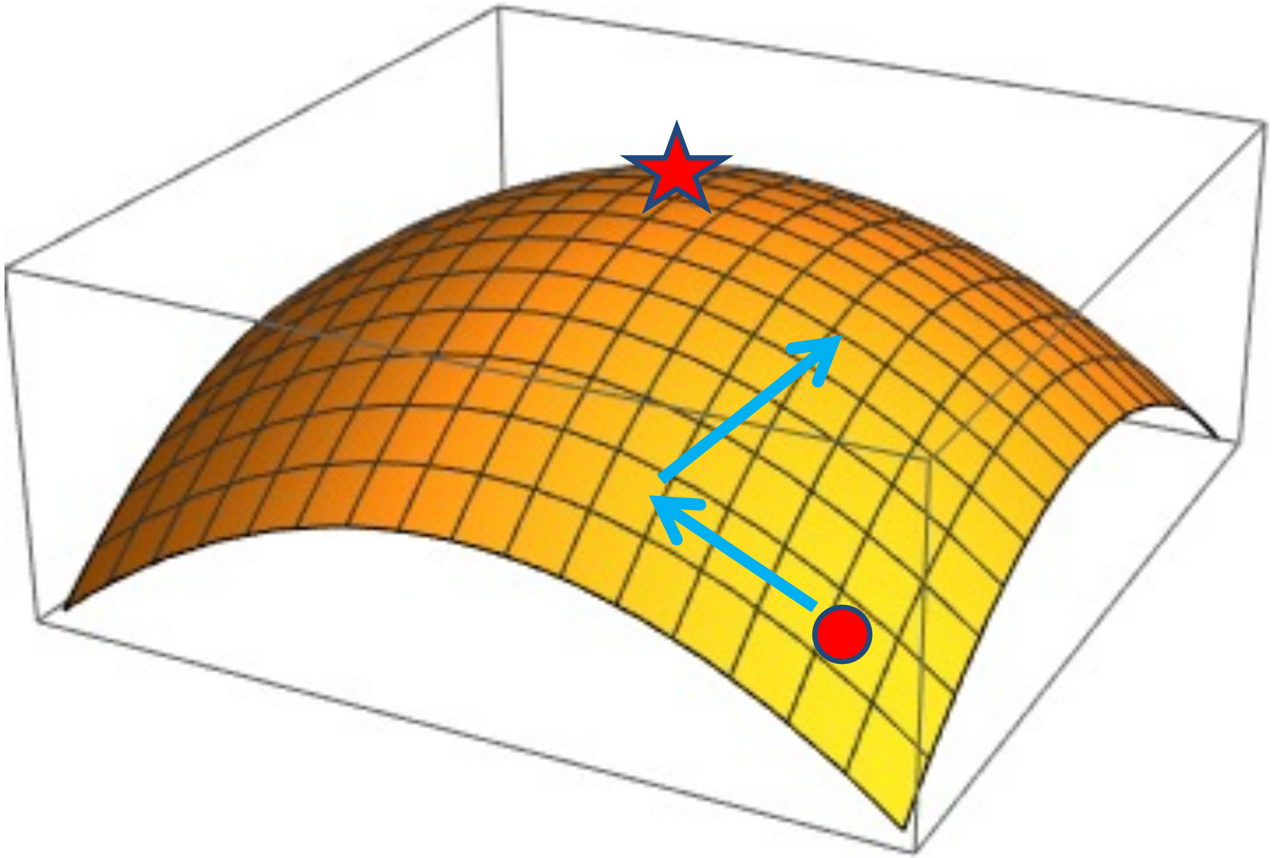
Gradient Ascent



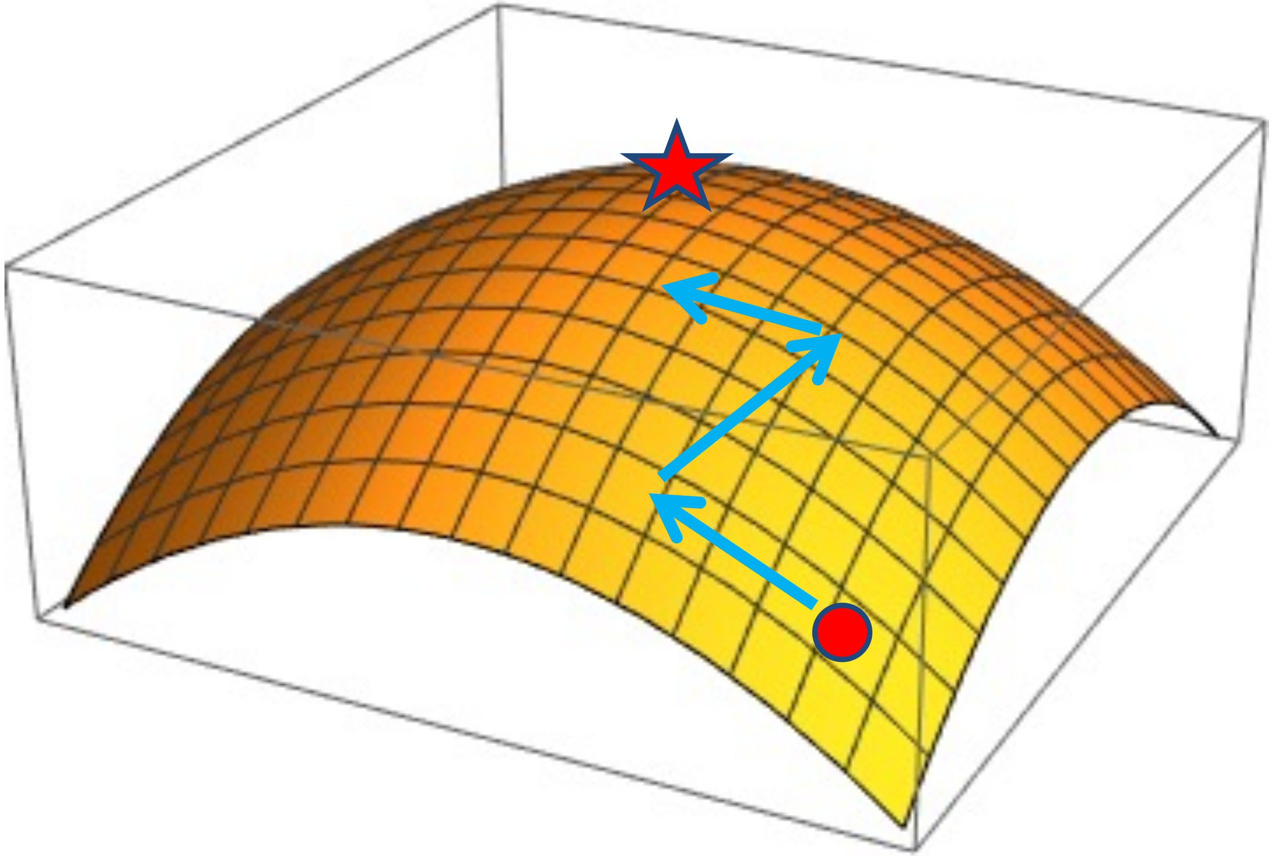
Gradient Ascent



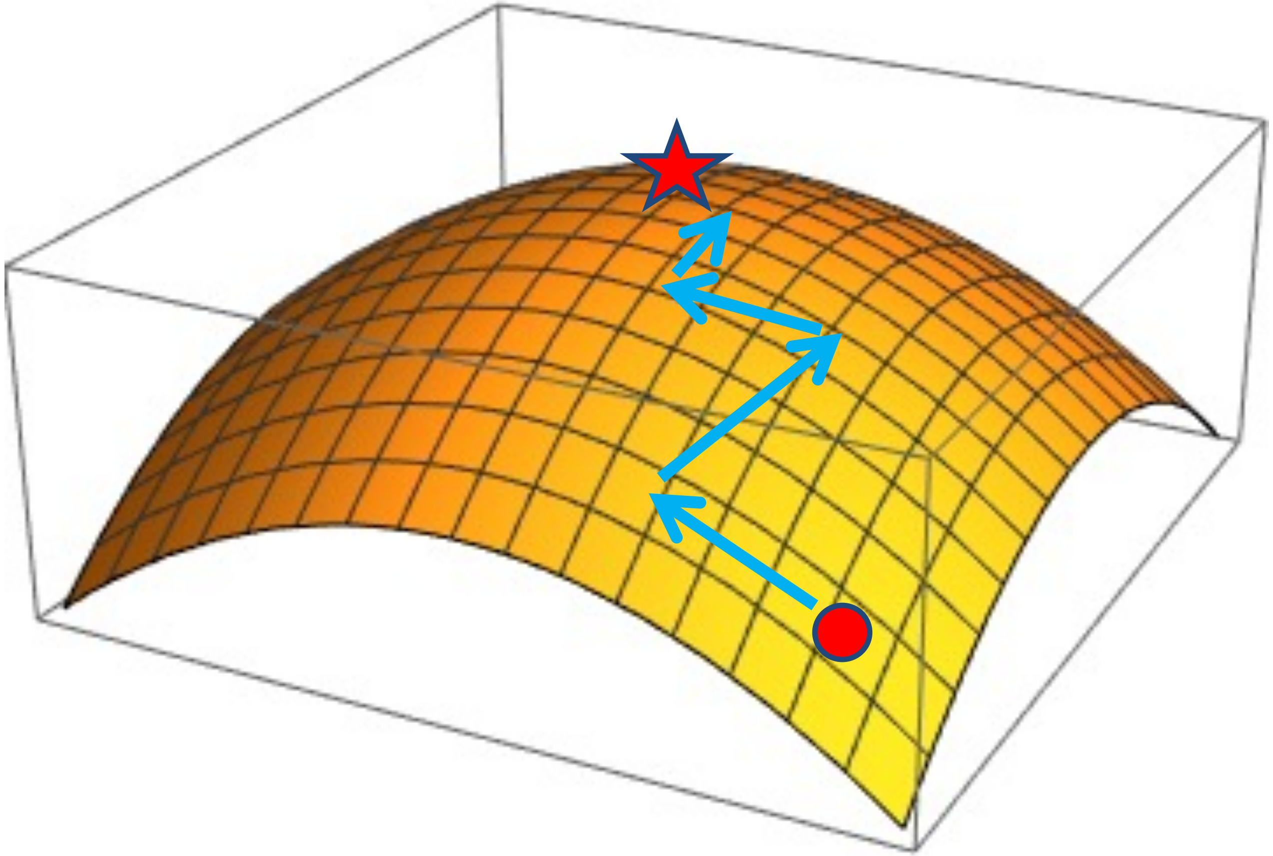
Gradient Ascent



Gradient Ascent



Gradient Ascent



What if you can’t find the roots? Follow the **gradient**

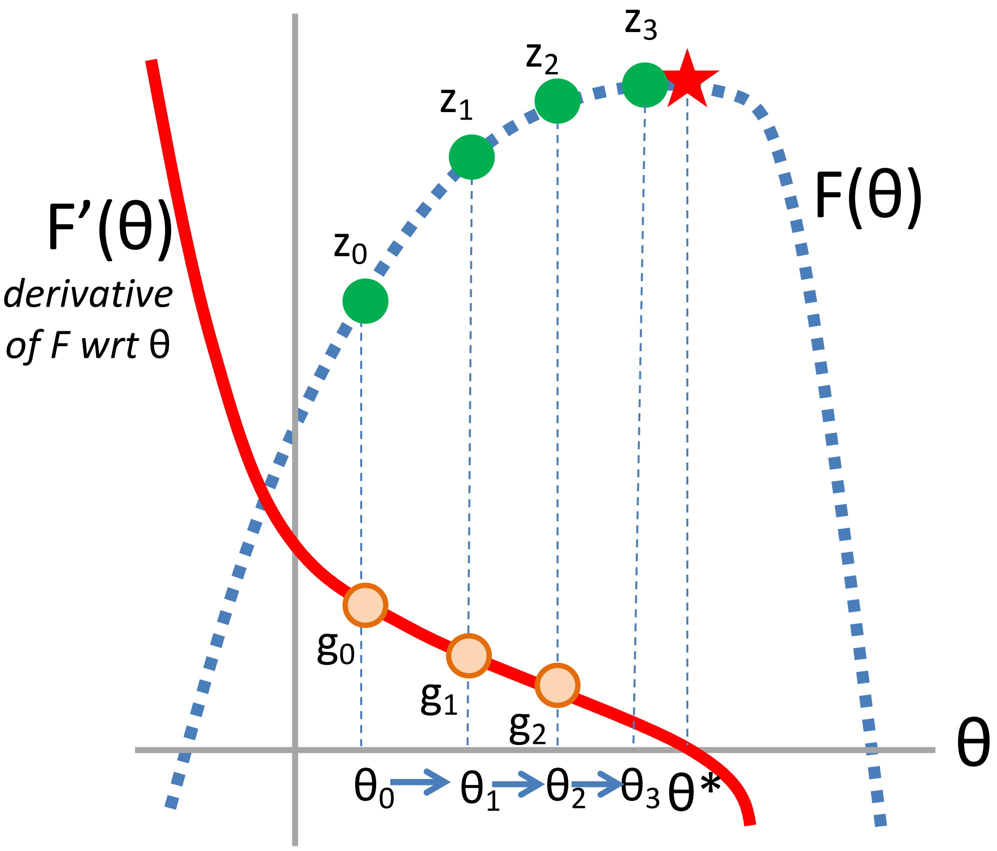
Set t = 0

Pick a starting value θ_t

Until converged:

1. Get value $z_t = F(\theta_t)$
2. Get **gradient** $g_t = F'(\theta_t)$
3. Get scaling factor ρ_t
4. Set $\theta_{t+1} = \theta_t + \rho_t * g_t$
5. Set t += 1

K-dimensional vectors



z_3
z_2
z_1
z_0
F’(θ)
derivative of F wrt θ
F(θ)
g_0
g_1
g_2
θ
$\theta_0 \rightarrow \theta_1 \rightarrow \theta_2 \rightarrow \theta_3\ \theta^*$

Outline

Maximum Entropy classifiers

Defining the model

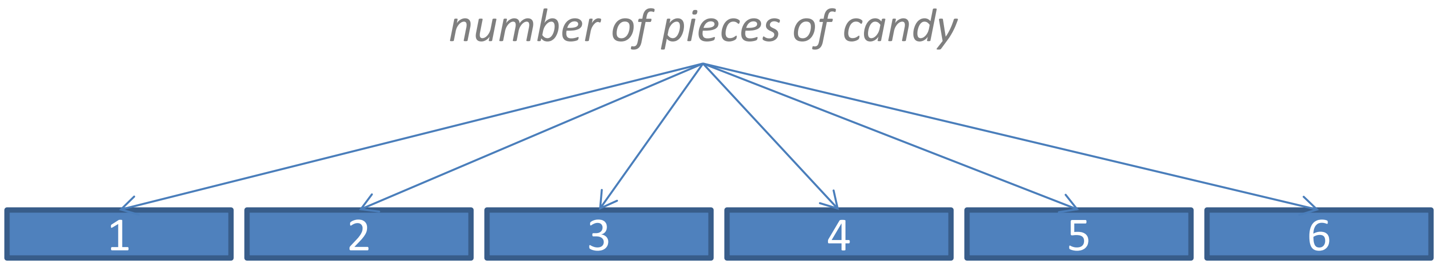
Defining the objective

Learning: Optimizing the objective

Math: gradient derivation (advanced)

Everything after this in this slide deck is "advanced" (not required, but *highly* recommended for any PhD or MS thesis student)

Expectation of a Random Variable



number of pieces of candy

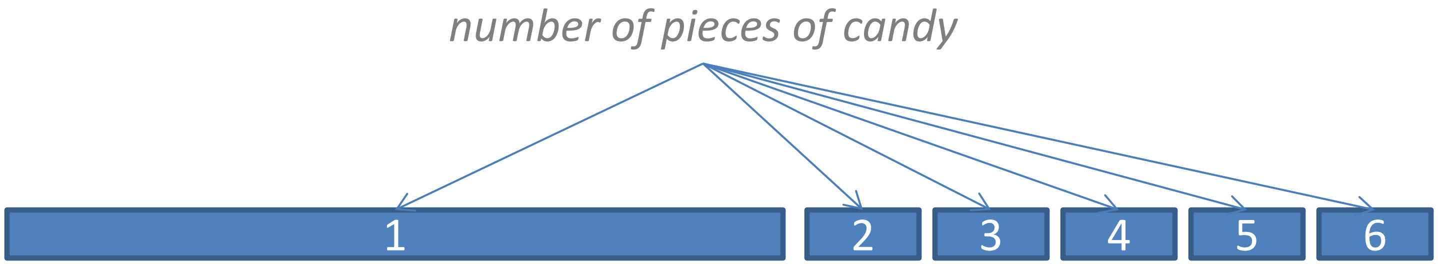
1 2 3 4 5 6

1/6 * 1 +
1/6 * 2 +
1/6 * 3 +
1/6 * 4 +
1/6 * 5 +
1/6 * 6

= 3.5

$$\mathbb{E}[X] = \sum_x x\, p(x)$$

Expectation of a Random Variable

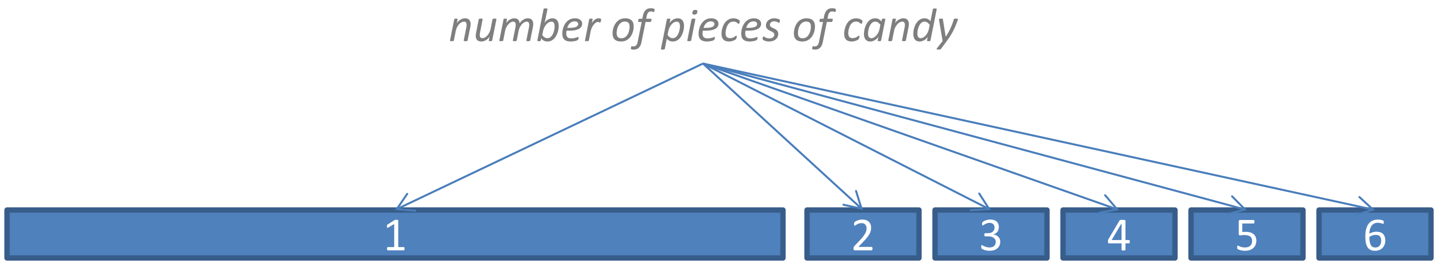


1/2 * 1 +
1/10 * 2 +
1/10 * 3 +
1/10 * 4 +
1/10 * 5 +
1/10 * 6

= 2.5

$$\mathbb{E}[X] = \sum_x x\, p(x)$$

Expectation of a Random Variable

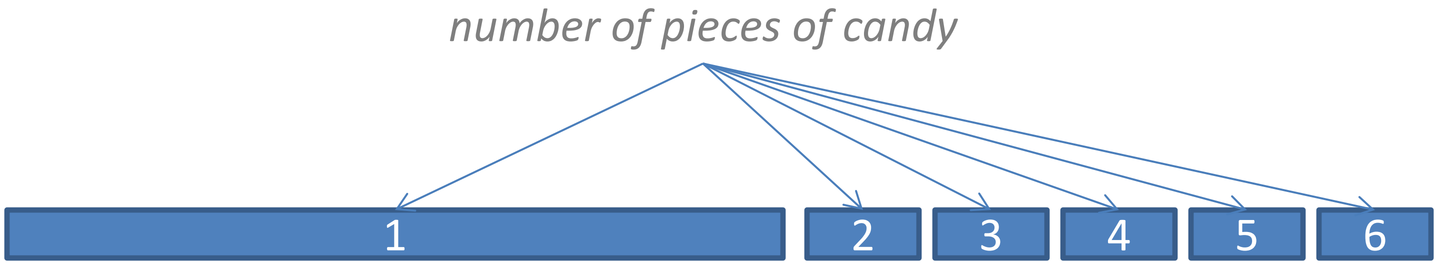


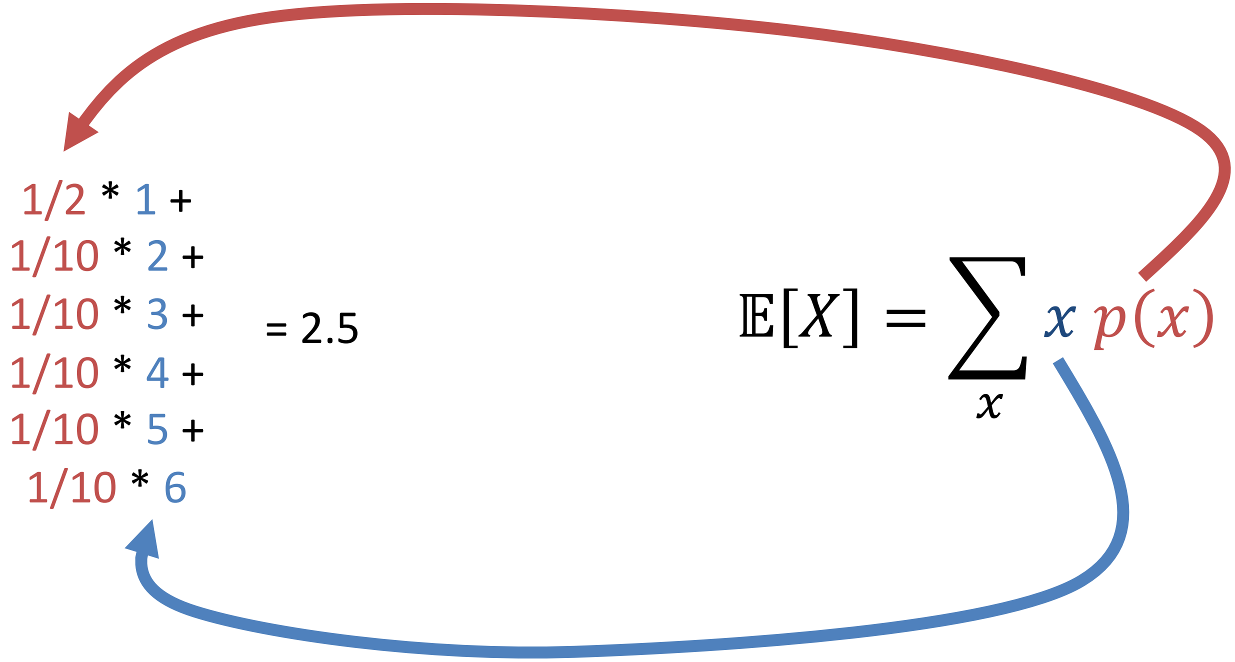
1/2 * 1 +
1/10 * 2 +
1/10 * 3 +
1/10 * 4 +
1/10 * 5 +
1/10 * 6

= 2.5

$$\mathbb{E}[X] = \sum_{x} x\, p(x)$$

Expectations Depend on a Probability Distribution





1/2 * 1 +
1/10 * 2 +
1/10 * 3 +
1/10 * 4 +
1/10 * 5 +
1/10 * 6

= 2.5

$$\mathbb{E}[X] = \sum_{x} x \, p(x)$$

Log-Likelihood Gradient

Each component for label *l* and feature *k* is the difference between:

Log-Likelihood Gradient

Each component for label *l* and feature *k* is the difference between:

the total value of feature f_k in the training data occurring with label *l*

$$\sum_i 1[y_i = l] f_k(x_i)$$

Log-Likelihood Gradient

Each component for label *l* and feature *k* is the difference between:

the total value of feature f_k in the training data occurring with label *l*

$$\sum_i 1[y_i = l] f_k(x_i)$$

and

the total value the current model p_θ *thinks* it computes for feature f_k with label *l*

$$\sum_i \mathbb{E}_{y' \sim p(y' | x_i)}[1[y' = l] f_k(x_i)]$$

"Moment Matching"

Log-Likelihood Gradient Derivation

$$\nabla_\theta F(\theta) = \nabla_\theta \sum_i \left[\theta_{y_i}^T f(x_i) - \log Z(x_i)\right]$$

Remember: Common Derivative Rules

$$\frac{d \exp x}{dx} = \exp x$$

$$\frac{df(x)g(x)}{dx} = \frac{df(x)}{dx} g(x) + \frac{dg(x)}{dx} f(x)$$

$$\frac{d \log x}{dx} = \frac{1}{x}$$

$$\frac{df(g(x))}{dx} = \frac{df(g(x))}{dg(x)} \frac{dg(x)}{dx}$$

Log-Likelihood Gradient Derivation

$$\nabla_\theta F(\theta) = \nabla_\theta \sum_i \left[\theta_{y_i}^T f(x_i) - \log Z(x_i)\right]$$

$$= \sum_i f(x_i) -$$

$$Z(x_i) = \sum_{y'} \exp(\theta_{y'} \cdot f(x_i))$$

Log-Likelihood Gradient Derivation

$$\nabla_\theta F(\theta) = \nabla_\theta \sum_i \left[\theta_{y_i}^T f(x_i) - \log Z(x_i)\right]$$

$$= \sum_i f(x_i) - \sum_i \sum_{y'} \underbrace{\overbrace{\frac{\exp\left(\theta_{y'}^T f(x_i)\right)}{Z(x_i)}}}_{\text{scalar } p(y' \mid x_i)} \underbrace{f(x_i)}_{\text{vector of functions}}$$

use the (calculus) chain rule

$$\frac{\partial}{\partial\theta} \log g(h(\theta)) = \left(\frac{\partial g}{\partial h(\theta)}\right)\left(\frac{\partial h}{\partial\theta}\right)$$

Log-Likelihood Gradient Derivation

$$\nabla_\theta F(\theta) = \nabla_\theta \sum_i \left[\theta_{y_i}^T f(x_i) - \log Z(x_i)\right]$$

$$= \sum_i f(x_i) - \sum_i \sum_{y'} \frac{\exp\left(\theta_{y'}^T f(x_i)\right)}{Z(x_i)} f(x_i)$$

Do we want these to *fully* match?

What does it mean if they do?

What if we have missing values in our data?

Gradient Optimization for Classifier $p(y \mid 📄)$

```
Set t = 0
Pick a starting value $\theta_t$
Until converged:
1. Get func. value $F(\theta_t)$
2. Get derivative $g_t = F'(\theta_t)$
3. Get scaling factor $\rho_t$
4. Set $\theta_{t+1} = \theta_t + \rho_t * g_t$
5. Set t += 1
```

$$\theta_y^T f(📄) - \log Z(📄)$$

$$\frac{\partial F}{\partial \theta_{k,y}} = f_{k,y}(📄) - \sum_{y'} f_{k,y'}(📄) p(y' | 📄)$$

Outline

Maximum Entropy classifiers

- Defining the model
- Defining the objective
- Learning: Optimizing the objective
- Math: gradient derivation (advanced)